

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LEONARDO DE MIRANDA BORBA

**Análise comparativa de dois modelos de
retransmissão adaptativa de vídeo para
múltiplos pontos em redes *unicast***

Trabalho de Graduação.

Prof. Dr. Valter Roesler
Orientador

Porto Alegre, dezembro de 2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. João César Netto

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço ao Valter Roesler que buscou, sempre, o aprimoramento dos seus orientandos.

Agradeço ao Ronaldo Hüsemann, que me auxiliou no Trabalho de Graduação I e foi importantíssimo para o aprendizado dos conceitos inerentes a codificação e escalabilidade de vídeo.

Agradeço aos meus colegas do grupo PRAV (Projetos em Áudio e Vídeo) pelo apoio e pelas contribuições ao trabalho. Em especial, agradeço ao Daniel Weber, por ter permitido que eu atribuísse maior prioridade às tarefas do Trabalho de Graduação, em detrimento de algumas tarefas da bolsa, e ao Leonardo Daronco, pelas contribuições com a tradução do Resumo e ajudas no âmbito do conhecimento.

Agradeço especialmente aos amigos Gabriel Pereira e Vanius Zapalowski pela leitura e revisão do texto.

E, por fim, um agradecimento especial para minha mãe, meu pai, minha irmã, minha tia Andréa e sua família e a todos os familiares e amigos, pois estes são os responsáveis por eu chegar até aqui.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Contexto histórico	13
1.2 Objetivo	14
1.3 Estrutura do documento	15
2 ESCALABILIDADE DE VÍDEO	16
2.1 Modelos de escalabilidade	18
2.1.1 Escalabilidade temporal	18
2.1.2 Escalabilidade espacial	19
2.1.3 Escalabilidade de qualidade	19
2.1.4 Escalabilidade híbrida	21
2.2 O padrão SVC do H.264	21
2.3 JSVM	22
2.4 Resumo	23
3 NETWORK ABSTRACTION LAYER (NAL) DO H.264	24
3.1 NAL Units	24
3.1.1 Tipos de <i>NAL Units</i>	25
3.1.2 Cabeçalho das <i>NAL Units</i> de vídeo escalável	27
3.2 Identificação de camadas	29
3.3 Resumo	29
4 MODELOS DE RETRANSMISSÃO DE VÍDEO ADAPTÁVEL NA REDE 30	
4.1 Transmissão de vídeo	30
4.2 Adaptação do vídeo às condições da rede	32
4.2.1 Transcodificador de vídeo	32
4.2.2 Roteador de vídeo escalável	33
4.3 H.323	33
4.3.1 MCU	34

4.4	Resumo	36
5	METODOLOGIA	37
5.1	Modelos avaliados	37
5.2	Testes propostos	38
5.2.1	Definição dos vídeos utilizados	38
5.2.2	Demanda de rede	40
5.2.3	Demanda computacional do retransmissor	40
5.2.4	Escalabilidade do processo	40
6	DESENVOLVIMENTO	41
6.1	Bibliotecas utilizadas	41
6.1.1	Bibliotecas do projeto IVA	41
6.1.2	Código do BitstreamExtractor	42
6.2	Desenvolvimento e implementação dos protótipos	43
6.2.1	BitstreamSender	44
6.2.2	BitstreamReceiver	44
6.2.3	BitstreamTranscoder	44
6.2.4	BitstreamScalableRouter	44
7	RESULTADOS	46
7.1	Codificação dos vídeos	46
7.2	Demanda de rede	48
7.2.1	Ambiente de testes	48
7.2.2	Resultados	49
7.3	Demanda computacional	52
7.3.1	Ambiente de testes	52
7.3.2	Resultados	52
7.4	Escalabilidade do processo	55
7.4.1	Ambiente de Testes	56
7.4.2	Resultados	56
8	CONCLUSÃO	58
	REFERÊNCIAS	60
	APÊNDICE A RESULTADOS DOS TESTES DE DEMANDA DE REDE	64
	APÊNDICE B RESULTADOS DOS TESTES DE DEMANDA COMPUTACIONAL	68

LISTA DE ABREVIATURAS E SIGLAS

AVC	Advanced Video Coding
CGS	Coarse-Grain Scalability
DCT	Discrete Cosine Transform
DID	Dependency ID
GOP	Group Of Pictures
IDC	Identification Code
IDR	Instantaneous Decoding Refresh
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPTV	IP Television
ITU-T	International Telecommunication Union
IVA	Sistema Interativo de Vídeo e Áudio
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
MC	Multipoint Controller
MCU	Multipoint Control Unit
MGS	Medium-Grain Scalability
MP	Multipoint Processor
MPEG	Moving Picture Experts Group
NAL	Network Abstraction Layer
PRID	Priority Identification
PSNR	Peak Signal-to-Noise Ratio
QID	Quality ID
RTP	Real-time Transport Protocol
SEI	Supplemental Enhancement Information
SNR	Signal-to-Noise Ratio

SVC	Scalable Video Coding
TID	Temporal ID
UDP	User Datagram Protocol
VBR	Variable Bitrate
VCL	Video Coding Layer

LISTA DE FIGURAS

Figura 2.1:	Vídeo de fluxo único	16
Figura 2.2:	Transmissão usando simulcast	17
Figura 2.3:	Transmissão usando vídeo escalável (HÜSEMANN, 2008)	17
Figura 2.4:	Exemplo de quadros I e P com tamanho de GOP igual a 3	18
Figura 2.5:	Método usado no H.262/MPEG-2	18
Figura 2.6:	Método para escalabilidade espacial	19
Figura 2.7:	Método para escalabilidade de qualidade por particionamento de dados.	20
Figura 2.8:	Escalabilidade híbrida tri-dimensional	21
Figura 3.1:	Primeiro byte do cabeçalho de uma <i>NAL Unit</i>	25
Figura 3.2:	Cabeçalho de uma <i>NAL Unit SVC</i>	28
Figura 4.1:	Transmissão em redes <i>multicast</i>	31
Figura 4.2:	Transmissão em redes <i>unicast</i>	31
Figura 4.3:	Retransmissor de vídeo	32
Figura 4.4:	Transcodificador de vídeo	33
Figura 4.5:	Roteador de vídeo escalável	34
Figura 4.6:	Formatos de vídeo conferência multi-pontos	35
Figura 7.1:	Ambiente para os testes de demanda de rede	48
Figura 7.2:	Demanda de rede nos três níveis do vídeo Ducks Take Off com resolução variando	50
Figura 7.3:	Demanda de rede nos três níveis do vídeo Ducks Take Off com qualidade variando	51
Figura 7.4:	Ambiente para os testes de demanda de rede	52
Figura 7.5:	Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Ducks Take Off com resolução variando	53
Figura 7.6:	Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Ducks Take Off com qualidade variando	54
Figura 7.7:	Ambiente para os testes escalabilidade do processo	56

LISTA DE TABELAS

Tabela 1.1:	Linha do tempo dos padrões de codificação	14
Tabela 2.1:	Exemplos de codificação escalável por <i>bitplanes</i>	20
Tabela 3.1:	Tipos de <i>NAL Unit</i> (ITU-T, 2010)	26
Tabela 7.1:	Definição dos níveis para o vídeo Ducks Take Off, com resolução variando	46
Tabela 7.2:	Definição dos níveis para o vídeo Ducks Take Off, com qualidade variando	46
Tabela 7.3:	Definição dos níveis para o vídeo Parkrun, com resolução variando	47
Tabela 7.4:	Definição dos níveis para o vídeo Parkrun, com qualidade variando	47
Tabela 7.5:	Definição dos níveis para o vídeo Mobcal, com resolução variando	47
Tabela 7.6:	Definição dos níveis para o vídeo Mobcal, com qualidade variando	47
Tabela 7.7:	Configurações das máquinas de teste	49
Tabela 7.8:	Acréscimo de taxa de bits inserido pelo Vídeo Escalável	52
Tabela 7.9:	Configurações das máquinas de teste	55
Tabela 7.10:	Configurações das máquinas de teste	57
Tabela 7.11:	Número de receptores suportados pelo retransmissor	57

RESUMO

Este trabalho de graduação apresenta, analisa e compara, com base em implementação e testes, dois modelos de retransmissão adaptativa de vídeo. O primeiro, o transcodificador de vídeo, tem como função recodificar os quadros recebidos segundo os parâmetros exigidos por cada um dos receptores. Já o roteador de vídeo escalável, apenas exclui pacotes que não são necessários à exibição do vídeo com uma determinada qualidade desejada.

O trabalho visa analisar criticamente, com base em testes, cada um dos pontos que diferenciam os dois métodos. São comparados, especificamente, a demanda de rede necessária para transmissão e recepção em ambos os métodos, a demanda de processamento para execução dos dois métodos e o número de receptores que podem ser associados a cada um dos retransmissores.

Palavras-chave: Escalabilidade de vídeo, Roteador de vídeo escalável, Transcodificador de vídeo, SVC, H.264, Codificação de vídeo.

Comparative analysis of two models of adaptive video retransmission to multiple points in unicast networks

ABSTRACT

This work presents, analyzes and compares, based on implementation and testing, two adaptive video retransmission models. The first one, a video transcoder, re-encodes the received frames to adapt them to the requirements of each receiver. The second model, a scalable video router, simply discards unnecessary packets to adapt the scalable video to the desired quality.

This work aims to analyze scientifically, based on tests, every aspect that differentiates the two methods. The aspects considered are the network bandwidth demanded for transmission and reception, the processing required to execute the method, and the number of receivers that can be associated to the transmitter.

Keywords: Video Scalability, Scalable Video Router, Video Transcoder, SVC, H.264, Video Coding.

1 INTRODUÇÃO

Com o avanço das tecnologias digitais, o uso de multimídia se tornou cada vez mais presente nas áreas de comunicação e entretenimento (FEIJÓO et al., 2007). Aplicações de comunicação interpessoal foram estendidas e, hoje em dia, é possível fazer reuniões virtuais com pessoas em diversos lugares do mundo, ter aulas a distância com interatividade ou, até mesmo, realizar acompanhamento médico remoto via telemedicina (MAS-SARENTI JR., 2006).

Para atender estas novas demandas, foram desenvolvidas diversas técnicas, visando melhorar as interfaces de comunicação e exibição multimídia pela inclusão de tecnologias cada vez mais sofisticadas. Particularmente, uma das mais importantes técnicas desenvolvidas é a chamada codificação de vídeo que tem por finalidade comprimir as informações necessárias para a representação de um vídeo, mantendo-se os atributos necessários para sua exibição. Com esses codificadores, pode-se reduzir significativamente as demandas de espaço de armazenamento e banda de transmissão para aplicações que envolvam sequências de vídeo (JACK, 2007).

Muitos padrões de codificação de vídeo foram definidos e alguns dos quais vêm sendo largamente utilizados até hoje (RICHARDSON, 2003). Diversos destes padrões de codificação encontram-se implementados na biblioteca FFMPEG¹, que apresenta código-fonte aberto.

Apesar dos diversos avanços presentes nos codificadores de vídeo, um problema ainda percebido para aplicações recentes de multimídia é a dificuldade de comportar, transparentemente, dispositivos com distintas características tecnológicas (capacidades de processamento, exibição e comunicação) dentro de um mesmo sistema de comunicação. Sob este aspecto torna-se importante encontrar métodos para satisfazer, de forma mais otimizada, as necessidades de cada dispositivo. Isto permite a um aparelho com determinadas limitações de resolução ou qualidade (aparelho celular por exemplo) receber um vídeo com baixa banda de rede, ao mesmo tempo em que uma televisão de alta-definição atende às suas aplicações utilizando banda maior (SCHWARZ; MARPE; WIEGAND, 2007).

Objetivando suprir as necessidades da arquitetura heterogênea da Internet, foram e estão sendo desenvolvidas uma série de soluções de vídeo adaptativo às condições de rede (GIROD et al., 2002).

As soluções adaptativas de vídeo, são divididas em duas fases. A primeira analisa as condições de rede entre o receptor e o emissor. A segunda, tratada neste trabalho, é a que adapta a codificação de vídeo às condições encontradas na fase de análise (GIROD et al., 2002).

Um modelo simples de adaptação às condições da rede, que funciona para o mo-

¹Disponível na página <http://ffmpeg.org>, no dia 7 de dezembro de 2010

delo de comunicação *unicast*, envolve a transcodificação de vídeo. Uma máquina fica responsável por receber, decodificar, adaptar o vídeo e recodificá-lo de acordo com as necessidades de cada terminal e, por fim, enviar aos receptores. Isto garante que cada terminal receba o vídeo da maneira que convier às suas necessidades mas, por ser um processo de retransmissão lento, envolvendo decodificação e codificação, este modelo se torna computacionalmente custoso (VETRO; CHRISTOPOULOS; SUN, 2003).

Outras possibilidades se baseiam na utilização de vídeo com codificação escalável. O vídeo escalável é dividido em camadas que podem ser usadas separadamente, respeitando as suas dependências. Isto dá margem a dois outros métodos de transmissão em redes heterogêneas: Um para transmissão *multicast* e outro para *unicast*.

Em redes *multicast*, separando as camadas do vídeo escalável em grupos *multicast* diferentes, é possível que cada terminal escolha quais camadas quer receber, simplesmente selecionando os grupos *multicast* com os quais deseja se conectar (TAN; ZAKHOR, 2001).

Utilizando um retransmissor, como no caso do transcodificador de vídeo, é possível que, em transmissões *unicast*, sejam passadas para cada receptor apenas as camadas necessárias; eliminando, portanto, o processo de recodificação presente no transcodificador (SOLER, 2006) e permitindo o uso da arquitetura atual da Internet em que o *multicast* não é garantido para todas as transmissões.

Prevendo o modelo atual da Internet, em que não se pode assumir que os diversos pontos da rede podem comunicar-se através de *multicast*, neste trabalho serão compreendidos os modelos de retransmissão adaptativa de vídeo através de redes *unicast*. Em especial serão tratados os modelos de transcodificação de vídeo e roteamento de vídeo escalável.

1.1 Contexto histórico

Desde os anos 80, foram definidos diversos padrões de codificação de vídeo, visando diminuir a quantidade de dados necessários para que um vídeo pudesse ser transmitido ou armazenado, evitando-se, ao máximo, a perda de qualidade. O primeiro grande passo para a definição de um padrão de codificação para uso massivo foi dado pela ITU-T, através do H.261, para uso em transmissões de vídeo. Seguindo-se a isto, e usando como base o próprio H.261, o grupo MPEG da ISO/IEC iniciou os trabalhos que culminaram no MPEG-1, para uso em armazenamento de vídeo, melhorando a eficiência da codificação, porém tomando maior demanda computacional (GHANDI; GHANBARI, 2004).

Advindo da necessidade de um padrão mais flexível, o grupo MPEG criou o MPEG-2 e, com a utilização em larga escala deste e pelo fato de a flexibilidade deste permitir melhor uso para telecomunicações, a ITU-T o adotou e modificou, passando a chamá-lo de H.262. Com isto o MPEG-2/H.262 se tornou o padrão de codificação mais utilizado no mundo, sendo o primeiro exemplo de sucesso da junção dos dois grupos (GHANBARI, 2003).

Posteriormente a isso os dois grupos lançaram projetos em separado: o H.263 e o MPEG-4 Visual. Munidos da mesma ideia que deu origem ao MPEG-2/H.262, os grupos novamente se uniram visando alcançar o mesmo sucesso, formaram então a JVT, e iniciaram o projeto H.26L, que visava alcançar codificadores que sobrepujassem os melhores codificadores atuais. Este padrão único ficaria conhecido como MPEG-4 versão 10 pela MPEG e como H.264 pela ITU-T (SULLIVAN; TOPIWALA; LUTHRA, 2004).

Uma necessidade que surgiu da heterogeneidade dos sistemas existentes, foi a criação de esquemas que permitissem escalabilidade de vídeo. Por causa disto, desde o MPEG-2

vêm-se desenvolvendo métodos que melhorem a eficiência dos codificadores para redes heterogêneas. Neste já se previam os três modelos de escalabilidade (temporal, espacial e de qualidade, que serão abordados na seção 2.1) (HASKELL; PURI; NETRAVALI, 1997). No codificador H.263 não foram definidos, diretamente, métodos para escalabilidade, embora isto tenha sido definido posteriormente no anexo O do protocolo que foi chamado de H.263+ (CÔTÉ et al., 1998). E ainda, na linha de evolução, houve definição de perfis do MPEG-4 Visual que suportavam escalabilidade, embora o perfil básico deste não suportasse o conceito (KOENEN, 2002).

O passo mais recente para definir um padrão de escalabilidade, entretanto, foi dado nos últimos anos através do H.264/SVC. A chamada de propostas para o SVC foi feita em 2003 pela MPEG e após a escolha do trabalho desenvolvido pelo Heinrich-Hertz-Institute de Berlim (SCHWARZ et al., 2004) como base para o projeto, o esquema foi aprimorado e finalmente aprovado em julho de 2007, como um adendo ao H.264/AVC (SCHWARZ; MARPE; WIEGAND, 2007).

Na tabela 1.1 é exibida a evolução dos codificadores de vídeo, bem como o suporte a escalabilidade representado por cada um deles; mostrando-se, através desta, quanto esforço tem sido investido em codificação de vídeo escalável nos últimos anos e a importância de se desenvolverem métodos de menor custo e igualmente eficientes para prover escalabilidade.

Tabela 1.1: Linha do tempo dos padrões de codificação

Padrão	Ano	Suporte a Escalabilidade
H.261	1989	Não
MPEG-1	1992	Não
H.262/MPEG-2	1994	Temporal, Espacial e de Qualidade
H.263	1996	Temporal, Espacial e de Qualidade (No Anexo O do H.263+)
MPEG-4	1998	Temporal, Espacial e de Qualidade (Em perfis complexos do padrão)
H.264/MPEG-4 AVC	2003	Temporal, Espacial e de Qualidade (No Anexo G, chamado de SVC)

1.2 Objetivo

No modelo de comunicação *unicast*, para que vários pontos da rede possam assistir a um mesmo vídeo é necessário que estes vários clientes se associem a um serviço, como um servidor ou um roteador, que repassa o vídeo a cada cliente. Como será necessária a retransmissão e como estamos tratando de redes heterogêneas, uma possibilidade interessante seria retransmitir o vídeo com os parâmetros definidos pelas condições de comunicação com os receptores, economizando banda nos casos necessários.

Quando o vídeo a ser retransmitido é codificado com o recurso da escalabilidade, o retransmissor, simplesmente, seleciona as camadas a serem retransmitidas para cada terminal, de acordo com suas necessidades.

Por outro lado, quando o vídeo não apresenta o recurso da escalabilidade, o modo de torná-lo adequado as necessidades de cada cliente é recodificar o vídeo segundo os novos

parâmetros de transmissão.

Neste trabalho, serão abordados estes dois métodos, sob ótica comparativa. Sobre-tudo, para cada um dos métodos, serão analisados:

- I) A banda de rede utilizada pelo retransmissor
- II) O consumo de CPU do retransmissor.
- III) A escalabilidade do retransmissor, ou seja, o número de receptores que é possível associar a este retransmissor.

O objetivo principal do presente trabalho de graduação é comparar soluções que facilitem a transmissão de vídeo em redes heterogêneas para transmissões *unicast*, incorporando a adaptação do vídeo às demandas de rede. Os critérios testados e analisados serão o transcodificador de vídeo e o roteador de vídeo escalável.

Devido à baixa disponibilidade de software livre e/ou gratuito que alcance as necessidades de tempo real para um codificador escalável, não serão feitas comparações objetivas que envolvam a codificação e decodificação executadas nos terminais.

1.3 Estrutura do documento

Para estabelecer um melhor conhecimento dos conceitos da escalabilidade de vídeo, importantes no presente trabalho, serão abordados, no Capítulo 2, alguns modelos de escalabilidade de vídeo e o protocolo H.264.

Para definir um vídeo H.264, tanto AVC quanto SVC, há duas camadas claramente separadas, o vídeo codificado e a *Network Abstraction Layer* que, entre outras coisas, separa adequadamente os quadros de vídeo codificado, informa os dados das camadas e armazena configurações do vídeo. No Capítulo 3, é descrito detalhadamente este protocolo, necessário para a identificação de camadas do SVC.

Em seguida, no Capítulo 4, são apresentados modelos de retransmissão adaptável de vídeo, analisando os prós e contras de cada método.

No Capítulo 5, por sua vez, são abordadas as metodologias aplicadas nos testes executados para a comparação.

O desenvolvimento dos aplicativos elaborados para os testes é, então, apresentado no Capítulo 6, mostrando bibliotecas, estruturas e o âmbito prático da implementação dos métodos.

Com base na execução dos testes, no Capítulo 7, são apresentados os resultados dos testes objetivos executados.

A partir destes resultados são apresentadas, no Capítulo 8, as conclusões referentes ao presente trabalho, incluindo uma pequena análise de trabalhos futuros.

2 ESCALABILIDADE DE VÍDEO

O modelo comumente empregado na transmissão digital de vídeo baseia-se no uso de um único fluxo de vídeo sendo enviado a um ou vários receptores. Neste sistema, cada receptor deve ser capaz de decodificar todo o fluxo que é enviado, que terá somente uma taxa de quadros e resolução. Este modelo funciona como mostrado na figura 2.1.

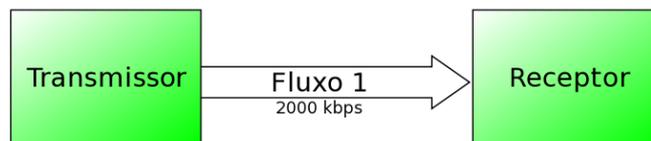


Figura 2.1: Vídeo de fluxo único

Entretanto, com a evolução das aplicações multimídia digitais, o modelo de fluxo único tornou-se um problema quando se trata de redes heterogêneas, ou seja, quando se tem aparatos dos mais diversos modelos e bandas de rede compartilhando a mesma transmissão, cada um com demanda de tráfego de rede diversa. Por exemplo, em IPTV, os mais diversos tipos de usuário teriam acesso a um mesmo fluxo de vídeo (HEI et al., 2007). Neste caso, transmitir vídeo com alta qualidade levaria os usuários com sistemas lentos e/ou baixa banda de rede a terem dificuldade para receber e/ou decodificar o vídeo em tempo real. Por outro lado, transmitir um vídeo de baixa qualidade subutilizaria os sistemas mais rápidos e com maior banda, fazendo com que estes usuários vissem vídeo de baixa qualidade apesar de disporem de excelentes recursos. O problema passa a ser, então, definir um padrão de transmissão em que cada receptor acompanha o vídeo com a qualidade que lhe é viável, sem subutilizar os sistemas mais poderosos e sem sobrecarregar os menos poderosos (HÜSEMANN, 2008).

Uma abordagem bastante simplista, adotada inicialmente, era transmitir diversos fluxos de vídeo, cada um com uma qualidade diversa. Ao receptor caberia escolher qual fluxo de vídeo seria o mais adequado para seus propósitos, recebendo-o e decodificando-o. O método, que tem seu funcionamento exibido na figura 2.2, recebeu o nome de simulcast (LIU; LI; ZHANG, 2006).

O problema latente ao método simulcast é o fato de ele codificar o mesmo vídeo sob diversas condições e não reaproveitar os dados entre os fluxos. A banda total necessária para o envio dos fluxos é, conseqüentemente, a soma das taxas de bits de cada vídeo codificado.

Para resolver este problema, foram estudados métodos para estabelecer relações entre os diversos fluxos de vídeo codificado e destes estudos emergiu a codificação escalável

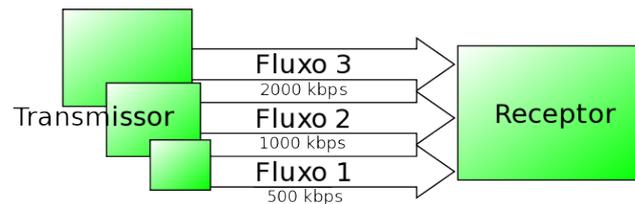


Figura 2.2: Transmissão usando simulcast

de vídeo.

Assim como no simulcast, na codificação escalável o vídeo é dividido em fluxos distintos, sendo que para a transmissão de um vídeo com maior taxa de quadros, maior resolução e/ou melhor qualidade serão codificados apenas os dados que o diferenciam do vídeo de qualidade, taxa de quadros e/ou resolução inferior. A camada com configurações mais simplórias, que não depende de nenhuma outra camada, é chamada de camada base. As outras camadas, por sua vez, são nomeadas camadas de enriquecimento, pois aumentam a qualidade do vídeo que será decodificado.

Por exemplo, se houver um vídeo sendo codificado sob três formatos diversos, para utilização em simulcast, e gerando taxas de bits de 500, 1000 e 2000 bits por segundo, a banda total necessária para o envio será a soma destas taxas de bits, ou seja: 3500 bits por segundo. Em um codificador escalável ideal, encontraria-se-ia a camada base com os mesmos 500 bits por segundo do simulcast. A primeira camada de enriquecimento, entretanto, teria, como taxa de bits de transmissão, a diferença entre a taxa da camada base e a taxa do vídeo enriquecido, ou seja, 500 bits por segundo. Esta subtração também aconteceria para a segunda camada de enriquecimento. Portanto, a taxa de bits total para o envio deste vídeo seria de 2000 bits por segundo. Este modelo é exemplificado na figura 2.3.

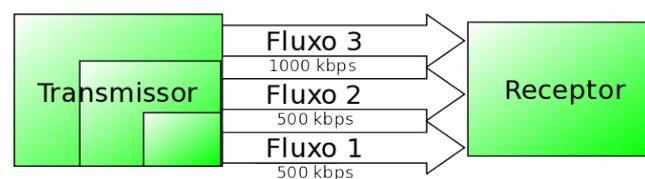


Figura 2.3: Transmissão usando vídeo escalável (HÜSEMANN, 2008)

Com os devidos indicadores de camada, os fluxos da escalabilidade podem ser unidos em um único fluxo, para poder ser transmitido por um único canal de comunicação (ITU-T, 2010).

Aproveitando-se dos fatores já presentes nos codificadores de vídeo e das noções conhecidas com relação à percepção humana do vídeo, pode-se estabelecer alguns modelos para o melhor aproveitamento dos dados presentes nas camadas inferiores de forma que as superiores necessitem menor taxa de transmissão.

Os principais modelos de escalabilidade conhecidos são discutidos a seguir.

2.1 Modelos de escalabilidade

2.1.1 Escalabilidade temporal

Um modelo de escalabilidade simples pode ser definido com base no fato de que um vídeo com determinada taxa de quadros pode ser facilmente dividido em vários fluxos com taxas de quadros menores, apenas selecionando os quadros. Desta forma, se obtém a chamada escalabilidade temporal, pois cada camada seria responsável por aumentar a taxa de quadros do vídeo recebido, ou seja, cada camada é responsável por uma parcela do tempo do vídeo exibido (CONKLIN; HEMAMI, 1999).

Um exemplo prático do uso de escalabilidade temporal se encontra no padrão H.262 / MPEG-2. Neste padrão estão definidos dois tipos de quadros em um vídeo, os quadros I, que são quadros que não dependem de outros para serem codificados, e os quadros P, que referenciam o último quadro I ou P, aproveitando os dados deste e codificando apenas a diferença para o mesmo, o que amplia a eficiência da codificação. Há, ainda, em codificadores mais modernos, como o H.264, os quadros B, que referenciam quadros anteriores e posteriores a ele, diminuindo, ainda mais, a taxa de bits necessários à transmissão (DARONCO, 2009).

Os tipos de quadro seguem um padrão repetitivo ao longo do vídeo, que é chamado de GOP (Group Of Pictures). Na figura 2.4 é mostrada uma sequência de quadros com tamanho de GOP igual a 3, usando apenas quadros I's e P's (HSIEH et al., 1995).

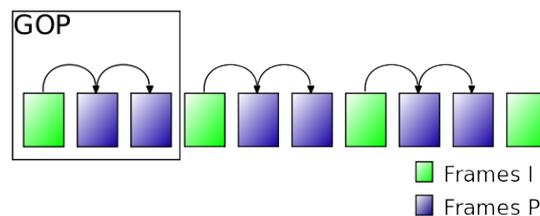


Figura 2.4: Exemplo de quadros I e P com tamanho de GOP igual a 3

Em um conceito escalável, os quadros I podem ser utilizados como camada-base, enquanto os quadros P, que dependem dos quadros I, podem ser usados como camadas de enriquecimento. Com isso os quadros ficam, simetricamente ou não, divididos entre as diversas camadas de forma bastante simples. Caberia ao receptor exibir somente os quadros I ou ambos os tipos de quadros, dependendo da sua capacidade de rede e/ou processamento (HASKELL; PURI; NETRAVALI, 1997). Um exemplo de divisão em 3 camadas de um vídeo codificado com GOP de tamanho 3 é mostrado na imagem 2.5.

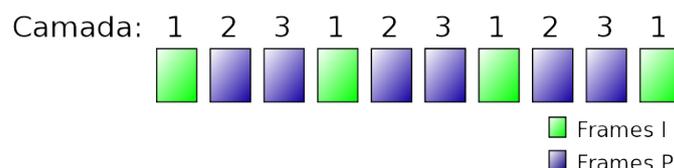


Figura 2.5: Método usado no H.262/MPEG-2

Este esquema, de simples implementação, tem contra ele o fato de que os quadros P e B são muito menores que os quadros I, o que faz com que se tenha uma camada base de tamanho muito elevado em relação às camadas de enriquecimento.

Ainda há, como ponto negativo do modelo, o fato de que os receptores que exibirem somente a camada base poderão ter frequência de quadros muito inferior àquela percebida pelo cérebro humano (WEBSTER et al., 1993), o que faria com que o usuário enxergasse o vídeo de forma descontinuada, perdendo a sensação de movimento habitual em vídeos.

Apesar destes pontos negativos, este modelo de escalabilidade tem a seu favor o fato de ser uma escalabilidade ideal, pois as camadas são disjuntas, ou seja, os dados das camadas de enriquecimento não sobrepõem os da camada base. Por este motivo, o vídeo escalável temporal tem a mesma taxa de bits do vídeo codificado sem escalabilidade equivalente, acrescentadas apenas informações relativas às camadas.

2.1.2 Escalabilidade espacial

Outra possibilidade para produzir escalabilidade é ter uma camada base com resolução baixa, mas ainda assim compreensível (isto se adaptaria perfeitamente a celulares, por exemplo, por não suportarem alta resolução e não terem, muitas vezes, rede de alta capacidade a disposição), e camadas de enriquecimento que adicionem os dados necessários ao aumento da resolução da imagem.

O modo mais simples de prover este tipo de escalabilidade, é fazer com que a camada base contenha a imagem de baixa resolução codificada convencionalmente e as outras camadas contenham apenas a diferença entre a imagem na nova resolução e a imagem produzida pelas camadas anteriores, como mostrado no exemplo da Figura 2.6. Isto, para imagens com pouco ruído, geraria camadas de enriquecimento com pouca variação de valores, o que as tornaria pequenas. Isto se deve ao fato de os codificadores atuais terem melhor desempenho quanto à taxa de bits do vídeo codificado quando há pouca variação entre os pixels.

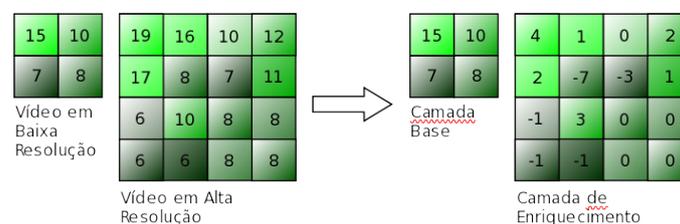


Figura 2.6: Método para escalabilidade espacial

Ao contrário do que acontece na escalabilidade temporal, na escalabilidade espacial o vídeo sempre terá a mesma frequência de exibição de quadros; pois, neste método, a taxa de quadros não é fator-chave na definição das camadas. Deve-se destacar, entretanto, que a escalabilidade espacial tem maior complexidade computacional, pois, para cada camada, é necessário efetuar um processo de codificação, além dos processos para estabelecer a relação entre a camada-base e as de enriquecimento.

2.1.3 Escalabilidade de qualidade

Nos dois modelos de escalabilidade anteriores, as camadas alteravam as métricas de tempo e espaço. De encontro a isso, existe um modelo de escalabilidade chamado de escalabilidade de qualidade, pois as camadas não afetam diretamente nenhuma das métricas do vídeo, mas modificam a percepção que se tem do mesmo. A camada-base terá a imagem com a qualidade mínima para se ter a ideia correta do vídeo, enquanto as camadas de enriquecimento aumentarão gradativamente a qualidade da imagem, permitindo a

exibição do vídeo com a percepção melhorada para o usuário (KONDI; ISHTIAQ; KAT-SAGGELOS, 1998).

Uma das técnicas mais simples de obter a escalabilidade de qualidade utiliza *bitplanes* para separar a imagem em camadas, ou seja, a representação binária dos pixels é utilizada para definição das camadas (LIN; HWANG; PEI, 2005).

Esta técnica se baseia no fato de que, cognitivamente, os bits mais significativos de cada cor dos pixels têm maior importância na informação das cores de uma imagem. Assim sendo, define-se uma camada-base composta pelos bits mais significativos de cada pixel, o que produz uma imagem com requisitos mínimos para compreensão, e as camadas de enriquecimento acrescentam qualidade às camadas anteriores pela inserção dos bits menos significativos (BRUNO, 2003), como mostrado na tabela 2.1.

Tabela 2.1: Exemplos de codificação escalável por *bitplanes*

Cor do Pixel	Camada 1	Camada 2	Camada 3
10010111	1001	01	11
11011100	1101	11	00
00001111	0000	11	11
10101010	1010	10	10

Um outro método, chamado de particionamento de dados, atua no domínio de frequência. Aproveitando a transformada DCT (*Discrete Cosine Transform*), aplicada comumente no processo de codificação, o codificador escalável seleciona os coeficientes correspondentes às frequências que serão incluídas em cada uma das camadas. A separação das camadas deve obedecer o fato de que os coeficientes no canto superior esquerdo da representação da DCT-2D se referem às menores frequências e os coeficientes do canto inferior direito representam as maiores frequências. Seguindo-se os coeficientes em ziguezague, é estabelecida a ordem entre as frequências dos coeficientes. Na camada base, portanto, são inseridos os coeficientes de menor frequência, fazendo com que a imagem perca parte do detalhamento, principalmente nas mudanças bruscas de cores entre pixels. Nas camadas de enriquecimento são inseridos os coeficientes relativos a outras frequências (DARONCO, 2009). Este modelo é apresentado na Figura 2.7 através de exemplo com duas camadas.

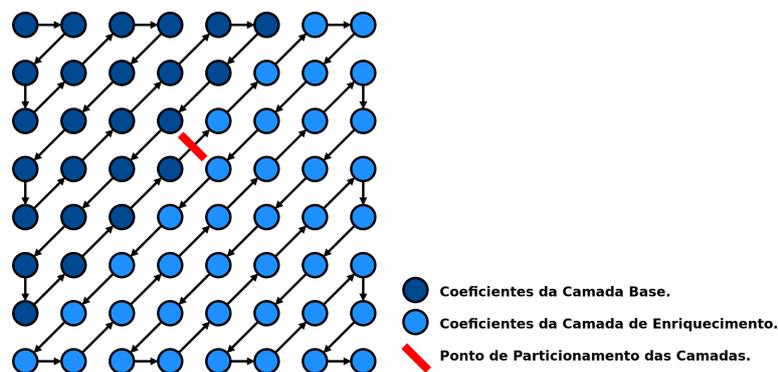


Figura 2.7: Método para escalabilidade de qualidade por particionamento de dados.

Há ainda o modelo de escalabilidade de qualidade em que a variação entre as camadas é feita no processo de quantização feito pelo decodificador. A camada base é quantizada de acordo com a precisão que se deseja obter da imagem com qualidade mínima. Nas camadas de enriquecimento, é codificada a diferença entre os níveis (ROBERS, 1997).

O principal ponto positivo dos modelos de escalabilidade por qualidade é o fato de as camadas não alterarem as métricas do vídeo. Além disso, o segundo e o terceiro modelos mostram que é possível desenvolver modelos de escalabilidade que não acrescentam grande complexidade computacional, aproveitando o processo de codificação convencional.

2.1.4 Escalabilidade híbrida

Ainda, há a possibilidade de, em modo multi-dimensional, juntar os métodos acima citados. A este tipo de junção de modelos dá-se o nome de escalabilidade híbrida. Com a escalabilidade híbrida é possível atender às diversas demandas de um sistema heterogêneo.

A figura 2.8 mostra um exemplo concreto de escalabilidade híbrida em que temos três níveis de escalabilidade espacial, três níveis de temporal e quatro de qualidade, totalizando 36 possibilidades de vídeo. Na imagem, os cubos demonstram os diversos modos de vídeo possíveis neste modelo e as setas indicam as camadas de enriquecimento necessárias para transicionar de um modo de vídeo para outro (SCHWARZ et al., 2005).

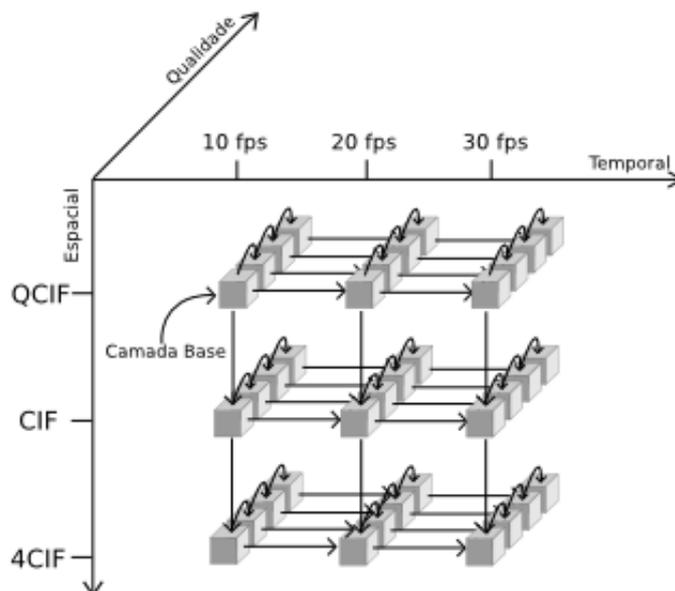


Figura 2.8: Escalabilidade híbrida tri-dimensional

2.2 O padrão SVC do H.264

O H.264/AVC (OSTERMANN et al., 2004) não foi desenvolvido com o propósito de ser um padrão de codificação escalável e, por isso, logo após o seu lançamento, o próprio grupo JVT iniciou os trabalhos para desenvolver um adendo a este protocolo, o anexo G do H.264, que passou a ser chamado de SVC (*Scalable Video Coding*). Este novo padrão escalável foi definido através do modelo proposto pelo Heinrich-Hertz-Institute do

Fraunhofer Institute de Berlim (SCHWARZ et al., 2004), escolhido entre muitos outros que participaram de um *call for proposals* feito no ano de 2003.

O SVC é o padrão escalável mais eficiente em uso atualmente, pois evita com maior eficácia as redundâncias entre camadas, através de métodos de predição inter-quadros. Ele engloba os três modelos de escalabilidade citados acima, suportando, portanto, escalabilidade híbrida (OELBAUM; SCHWARZ; WIEN, 2007).

Os modelos do H.264/SVC para escalabilidades temporal e espacial são muito semelhantes aos modelos mostrados nas subseções 2.1.1 e 2.1.2, acrescidos alguns fatores que visam melhorar a eficiência da codificação como o de quadros B hierárquicos na escalabilidade temporal (SCHWARZ; MARPE; WIEGAND, 2006) e o uso de predição inter camadas na escalabilidade espacial (SCHWARZ; MARPE; WIEGAND, 2007).

Há dois modos de escalabilidade de qualidade no H.264/SVC, o primeiro é um tipo de CGS (*Coarse-Grain Scalability*), que é muito semelhante ao modelo aplicado na escalabilidade espacial, excetuando-se o fato de que a resolução é mantida entre as camadas e a subtração é efetuada com base em camadas com parâmetros de quantização distintas, sendo inclusive aproveitados métodos de predição inter camadas (LI; LI; WEN, 2006). O segundo método, conhecido como MGS (*Medium-Grain Scalability*) permite maior flexibilidade pois permite que dentro de cada nível de escalabilidade CGS ou espacial existam diferentes quantidades de camadas MGS, aumentando as possibilidades de adaptação do vídeo (MANSOUR et al., 2009).

2.3 JSVM

JSVM¹ é um acrônimo para *Joint Scalable Video Model* e é o nome do software de referência para o padrão SVC do H.264.

A JSVM é um software livre desenvolvido na linguagem de programação C++ com objetivos didáticos e não tem, portanto, o objetivo de ser utilizado em ambientes reais de transmissão de vídeo. Devido a esta escolha pela simplicidade, em detrimento da otimização, os codificadores e decodificadores não utilizam, por exemplo, recursos de *multithreading*.

A codificação e decodificação na JSVM são, portanto, processos lentos e enfadonhos e sequer se aproximam de suprir as necessidades de um sistema de tempo-real.

A JSVM apresenta um conjunto de ferramentas e aplicativos de teste baseados em sua própria biblioteca de codificação e decodificação. Os mais importantes são:

- **H264AVCEncoderLibTestStatic** - Aplicativo desenvolvido para testes da biblioteca de referência de codificação. Recebe como entrada um arquivo de vídeo com luminância e croma codificados em formato YUV e um arquivo com as configurações do vídeo (em especial a definição das camadas). Retorna o vídeo codificado no padrão SVC do H.264.
- **H264AVCDecoderLibTestStatic** - Decodifica vídeo em formato SVC do H.264 e gera, como saída, um arquivo de vídeo com luminância e croma do vídeo em formato YUV.
- **BitStreamExtractorStatic** - Dados um arquivo de vídeo codificado através do SVC do H.264 e as configurações do vídeo requeridas, este programa extrai as

¹Disponível na página http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm, acessada no dia 7 de dezembro de 2010

camadas que não são necessárias para a exibição na qualidade desejada. Este procedimento gera um segundo arquivo apenas com as camadas necessárias do SVC.

- **DownConvertStatic** - Faz conversões em vídeos YUV, diminuindo largura, altura e/ou taxa de quadros.
- **PSNRStatic** - Dado um vídeo em YUV original e o vídeo em YUV após o processo de codificação e decodificação, calcula o PSNR (*Peak Signal-to-Noise Ratio*, definido na seção 5.2.1) correspondente.

Com estas ferramentas, é possível executar todas as operações com vídeos armazenados em arquivo que são exigidas ao longo do trabalho, como codificação, decodificação e cálculo do PSNR do vídeo.

2.4 Resumo

Este capítulo apresentou a escalabilidade de vídeo, bem como seus principais modelos: Escalabilidade Temporal, Espacial, de Qualidade e Híbrida.

A escalabilidade de vídeo é um processo pelo qual fluxos de vídeo com diferentes qualidade, resolução e/ou taxa de quadros podem ser codificados reaproveitando dados de um fluxo para outro e evitando a redundância de dados.

O principal padrão que suporta escalabilidade é o SVC do H.264 que disponibiliza escalabilidade híbrida como um anexo ao H.264/AVC. Este padrão pode ser codificado e decodificado utilizando a JSVM, o software de referência do H.264/SVC. Tanto o SVC quanto as demais partes do vídeo codificado do H.264 são encapsuladas pela *Network Abstraction Layer* do H.264, descrita no próximo capítulo.

3 NETWORK ABSTRACTION LAYER (NAL) DO H.264

No protocolo H.264 há duas camadas importantes para a correta codificação e transmissão de vídeo. A primeira diz respeito aos dados relativos à codificação e decodificação do vídeo, chamado de VCL (*Video Coding Layer*). No entanto, para que o H.264 esteja perfeitamente adaptado aos mais diversos modelos de transmissão, também existe uma camada de abstração de rede (NAL) (WIEGAND et al., 2003). A NAL provê informações referentes ao vídeo que concernem ao processo de codificação, decodificação e análise do vídeo, incluindo conjunto de parâmetros e indicadores de camadas de escalabilidade.

A NAL foi projetado especificamente com o propósito de ser amigável aos mais diversos protocolos de rede para transmissão de vídeo, facilitando o mapeamento da VCL do H.264 para os mais diversos canais de transmissão.

Embora tenha o nome de *Network Abstraction Layer*, a NAL não é utilizada somente para transmissão em redes, podendo ser utilizado também para armazenamento. A NAL é facilmente adaptável a, por exemplo:

- Protocolo RTP/UDP/IP (WENGER, 2003).
- Mídias de armazenamento (AMON; RATHGEN; SINGER, 2007).
- A família H.32x de recomendações da ITU-T, em protocolos como H.320 e H.324M (OSTERMANN et al., 2004).

A NAL que encapsula o vídeo H.264 é, na verdade, uma *stream* de *NAL Units*, podendo, estas, serem encapsuladas, das mais diversas formas, nos protocolos de transmissão ou armazenamento.

Inicialmente, o projeto da NAL não incluía vídeo escalável, sendo esta incluída já nas primeiras versões do H.264. Previa, entretanto, o acréscimo de novos tipos e cabeçalhos para novas *NAL Units*. Com a inclusão do Anexo G, referente ao SVC, a NAL foi atualizada para incorporar as novas demandas de vídeo escalável (ITU-T, 2010).

3.1 NAL Units

Os vídeos codificados em H.264, utilizando a *Network Abstraction Layer*, são divididos em *NAL Units*. Cada *NAL Unit* é iniciada por um cabeçalho que define, entre outras coisas, seu tipo e seu tamanho.

As *NAL Units* são claramente separadas em três classes:

- **Parameter Sets** - Definem conjuntos de parâmetros necessários à codificação do vídeo. Podem incluir: resolução do vídeo, parâmetros de quantização, *flags* de

controle do codificador e outros parâmetros que são importantes para que, posteriormente, o vídeo seja decodificado corretamente.

- **Slices** - Subconjunto de um quadro H.264 codificado. Na prática, tem um cabeçalho com dados e *flags* sobre aquele quadro, como se é um quadro I, P ou B; subsequente a isto, são encapsulados os dados codificados referentes àquele quadro.
- **Supplemental enhancement information** - Informações que não são necessárias ao processo de codificação e decodificação, mas que são importantes para, por exemplo, a correta bufferização do vídeo.

3.1.1 Tipos de NAL Units

O primeiro byte do cabeçalho de uma *NAL Unit*, que é mostrado na Figura 3.1, apresenta 3 informações, a saber: O primeiro bit deve ser preenchido com o valor zero. O segundo e terceiro bits indicam se a *NAL Unit* atual é (valor diferente de 0) ou não (valor igual a zero) uma imagem de referência, ou seja, se ela é utilizada para reconstruir imagens de referência. Os últimos 5 bits deste byte especificam o tipo da *NAL Unit*.

0	1	2	3	4	5	6	7
F	NRI		Type				

Figura 3.1: Primeiro byte do cabeçalho de uma *NAL Unit*

Por ser composto por 5 bits, são possíveis 32 tipos de *NAL Units*. Os tipos são enumerados na tabela 3.1.

3.1.1.1 Tipos 1 a 5 - Parte codificada de uma imagem não-IDR ou IDR

Os elementos da VCL são inseridos na *Network Abstraction Layer* em *slices*, podendo ser estes, parte ou o todo de um quadro H.264/AVC. Os tipos de *NAL Unit* relativos aos *slices* de vídeo são os tipos de 1 até 5.

No VCL estão presentes as chamadas imagens IDR (*Instantaneous Decoding Refresh*), que são utilizadas, geralmente, no início das sequências de vídeo, indicando uma atualização do decodificador. Os dados do decodificador são, quando utilizada uma imagem IDR, reinicializados e há garantia de que nenhum quadro recebido posteriormente utilizará, como referência, quadros anteriores à imagem IDR. Excetuando-se o primeiro quadro, os demais são, geralmente, Non-IDR.

Os tipos 1 e 5 encapsulam quadros inteiros de vídeo H.264/AVC, sendo separados em dois tipos: Non-IDR *Slice* e IDR *Slice*. Por sua vez, os tipos 2, 3 e 4 compreendem três categorias de partição de dados de um quadro, separados de forma semelhante ao método de escalabilidade apresentado na seção 2.1.3, que permite maior resiliência a erros, pois *slices* que não chegam ao decodificador apenas eliminam algumas frequências do vídeo, não o vídeo por completo (CONNIE et al., 2008).

Os tipos 2, 3 e 4, com partição de dados, não podem ser utilizados em vídeos adequados ao Anexo G (SVC) e não podem ser imagens IDR (ITU-T, 2010).

3.1.1.2 Tipo 6 - SEI (Supplemental Enhancement Information)

Alguns dados para, por exemplo, bufferização do vídeo são inseridos na *Network Abstraction Layer* através das *NAL Units* chamadas de SEI (Supplemental Enhancement

Tabela 3.1: Tipos de *NAL Unit* (ITU-T, 2010)

Tipo	Descrição
0	Não especificado
1	<i>Slice</i> codificado de uma imagem não-IDR sem partições
2	Partição A de dados de um <i>slice</i> codificado
3	Partição B de dados de um <i>slice</i> codificado
4	Partição C de dados de um <i>slice</i> codificado
5	<i>Slice</i> codificado de uma imagem IDR sem partições
6	<i>Supplemental Enhancement Information</i> (SEI)
7	Parâmetros de sequência
8	Parâmetros de imagem
9	<i>Access unit delimiter</i>
10	Fim da sequência
11	Fim do <i>stream</i>
12	<i>Filler data</i>
13	Extensão dos parâmetros de sequência
14	<i>Prefix NAL Unit</i>
15	Parâmetros de sequência para um subconjunto
16..18	Reservados
19	<i>Slice</i> de uma imagem auxiliar codificada.
20	<i>Slice</i> codificado utilizando SVC
21..23	Reservados
24..31	Não especificados

Information). Estas *NAL Units* não são necessárias para a codificação correta do vídeo sendo, no entanto, de extrema necessidade para a correta transmissão do vídeo.

3.1.1.3 Tipos 7 e 8 - *Sequence e picture parameter sets*

NAL Units que contêm informações que raramente mudam e referem-se à codificação de muitos quadros são chamadas de *parameter sets*.

Quando um *parameter set* tem informações referentes a uma sequência de vídeo, com parâmetros que dizem respeito à relação entre estes quadros, ele será chamado de *Sequence Parameter Set*. Entre os parâmetros mais importantes do *Sequence Parameter Set*, descritos na seção 7.4.2.1 de (ITU-T, 2010), podem-se destacar:

- *Sequence Parameter Set ID* - O ID que será referenciado pelas *Picture Parameter Set's* para associar os dois *parameter sets*.
- *Profile IDC* - Indica o perfil ao qual o *Sequence Parameter Set* se adapta.
- *Level IDC* - Indica o nível do perfil ao qual o *Sequence Parameter Set* se adapta.

Quando se referir a somente um ou a um conjunto de quadros de forma individual, com informações que são pertinentes à decodificação de cada um deles, será chamado de *Picture Parameter Set*. Na seção 7.4.2.2 de (ITU-T, 2010), podem-se encontrar as descrições dos parâmetros presentes em um *Picture Parameter Set*. O *Picture Parameter Set* referencia um *Sequence Parameter Set* de quem incorpora parte das configurações.

3.1.1.4 Tipo 14 - Prefix NAL Unit

Como as camadas-base do vídeo com escalabilidade espacial ou de qualidade devem apresentar dados suficientes para que possam ser decodificadas e, assim, apresentar vídeo de qualidade inferior, os quadros codificados para fazer parte da camada-base apresentam as mesmas propriedades do vídeo não escalável e podem ser inseridas, inclusive, nas mesmas *NAL Units*, os *slices* de quadro, que têm os tipos 1 e 5. Para identificar os ID's de camadas e outros parâmetros, todavia, é necessário prefixar estas *Units* com outra *NAL Unit*, chamada de Prefix Nal Unit (WANG et al., 2007).

A *NAL Unit* de tipo 1 ou 5, que sucede a prefix *NAL Unit*, é chamada de *NAL Unit* associada e, para efeitos de análise da NAL, a Prefix *NAL Unit* é considerada parte da *NAL Unit* associada, ou seja, a exclusão de uma implica a exclusão da outra (ITU-T, 2010).

Os cabeçalhos da Prefix *NAL Unit* têm campos para indicar separadamente o nível da camada de escalabilidade em cada uma das três possíveis formas de escalabilidade que o H.264/SVC apresenta.

3.1.1.5 Tipo 15 - Subset sequence parameter set

Este *parameter set* é utilizado para passar parâmetros de configuração a algum subconjunto de camadas-base ou de enriquecimento de um vídeo escalável. O *subset sequence parameter set* referencia um *sequence parameter set*, utilizando parâmetros de configuração passados através deste e acrescentando as informações específicas do subconjunto de camadas afetadas por ele (WANG et al., 2007).

3.1.1.6 Tipo 20 - Slice para codificação de camadas de enriquecimento do SVC

As camadas de enriquecimento do SVC, ao contrário das camadas-base, não podem ser codificadas segundo o padrão AVC básico, pois compreendem a diferença entre o vídeo produzido pelas camadas inferiores e o vídeo que se deseja produzir com a camada atual, ou seja, um processo de codificação e decodificação distinto das camadas-base (WANG et al., 2007).

A necessidade de um novo tipo de codificação originou a necessidade de identificação de um novo tipo de *slice* na camada de abstração de rede. A alternativa utilizada, para não se perder a compatibilidade com o H.264/AVC não escalável, foi adicionar uma nova *NAL Unit* que encapsula a VCL das camadas de enriquecimento, a *NAL Unit* de tipo 20.

Como, ao contrário do caso da camada-base, foi criada uma nova *NAL Unit* para a VCL, não é necessária a presença de uma Prefix *NAL Unit* para identificar e parametrizar a camada. O cabeçalho de identificação de camadas, idêntico ao da Prefix *NAL Unit*, é adicionado diretamente nas *NAL Units* de *slices* de vídeo escalável (ITU-T, 2010).

3.1.2 Cabeçalho das *NAL Units* de vídeo escalável

Os cabeçalhos das *NAL Units* do H.264/AVC, apresentam apenas os 8 bits ilustrados na Figura 3.1. Estes bits são suficientes para a identificação de *NAL Units* de vídeos não escaláveis. Com a introdução do vídeo escalável e, conseqüentemente, a necessidade de identificar as camadas, foram inseridos mais 3 bytes no cabeçalho das *NAL Units* 14 e 20, que encabeçam os *slices* de vídeo escalável. A formação dos 4 bytes é descrita na Figura 3.2 e detalhada em seguida, como mostrado em (WANG et al., 2007).

- **F** - Este é um bit reservado com valor 0. A modificação do valor para 1 constitui um erro semântico na recomendação H.264 e assim deve ser considerado pelas implementações.

0									1									2									3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
F	NRI	Type			R	I	PRID			N	DID		QID			TID		U	D	O	RR										

Figura 3.2: Cabeçalho de uma *NAL Unit SVC*

- **NRI** - NAL Reference IDC. Em *slices* de vídeo, esta *flag* indica se o quadro é um IDC (valor igual a 0) ou não (valor diferente de 0). Quanto maior o valor desta *flag*, maior a necessidade visual da correspondente *NAL Unit*. SEI's por exemplo, que não tem importância na reconstrução das imagens, têm valor zero, assim como imagens que não são referência para outras. Os quadros de referência e os *parameter sets*, por sua vez, têm valor máximo, pelo fato de serem importantíssimos para a correta decodificação.
- **Type** - Indica o Tipo da *NAL Unit*, como mostrado anteriormente.
- **R e RR** - Estes são bits reservados para futuras extensões.
- **I** - Indica se o quadro é um IDR (*Instantaneous Decoding Refresh*). Os quadros IDR são obrigatoriamente os primeiros em um vídeo. A utilização de um quadro IDR indica um reinício do decodificador e também significa que nenhum quadro precedente será utilizado como referência para quadros futuros.
- **PRID** - Identificador de prioridade da *NAL Unit*. Em escalabilidade híbrida, não é possível estabelecer um caminho de adaptação. Por exemplo, se queremos um vídeo CIF de 30 quadros por segundo (FPS) e temos, como camada-base, um vídeo QCIF de 15 FPS, não é possível estabelecer se a ordem de aplicação das camadas de enriquecimento é ampliar a taxa de quadros com uma camada de escalabilidade temporal e depois ampliar a imagem com uma de escalabilidade espacial ou vice-versa. Como esta ordem é importante na hora de definir quais camadas serão enviadas em um vídeo adaptável, foi definido um novo campo para especificar a prioridade do quadro, sendo que menores valores indicam maiores prioridades.
- **N** - Indica se pode ser usada predição inter-camadas a partir deste *slice*.
- **DID** - As camadas de escalabilidade espacial e CGS têm processo de codificação muito semelhante, por este motivo, evitando redundâncias, as duas compõem um mesmo modelo de escalabilidade. O DID (*Dependency ID*) é, portanto, o ID da camada CGS ou espacial correspondente. Um ID igual a 0 indica uma camada-base enquanto as camadas de enriquecimento, com valor diferente de zero, são ordenadas segundo sua prioridade.
- **QID** - A escalabilidade MGS é responsável por prover mais níveis de granularidade por qualidade. Para isso, o modelo é descrito nos cabeçalhos de *NAL Unit* no campo QID (*Quality ID*) com 4 bits, o que possibilita 16 camadas, 8 a mais do que os outros dois modelos de escalabilidade.
- **TID** - Na hierarquia de escalabilidade temporal, também é necessário identificar a que camada cada quadro pertence, por isso é estabelecido o TID (*Temporal ID*).

- **U** - Indica se o quadro corrente será utilizado como imagem de referência para imagens futuras. Como o decodificador incorpora um buffer de quadros para referência, é necessário estabelecer um limite de quadros para referência que limite o tamanho do buffer. Para evitar que os quadros de referência ultrapassem este limite, foi inserida esta *flag*, que permite ao codificador indicar os quadros que serão ou não utilizados como quadro de referência.
- **D** - Se a camada atual não é importante para a decodificação das camadas superiores, ela deve ser posicionado como descartável (D com valor igual a 1), caso contrário, D deve ser posicionado como 0.
- **O** - Quando este bit está posicionado, o quadro será retornado pelo decodificador após a decodificação, caso contrário ele apenas será decodificado.

3.2 Identificação de camadas

A identificação de uma camada é feita pelos ID's: DID (para escalabilidade espacial e CGS), QID (para escalabilidade MGS), TID (para escalabilidade temporal) e PRID (para indicar a prioridade da camada).

Como as prefix *NAL Units* encabeçam camadas-base, deve-se garantir que tanto o QID, quanto o DID tenham valor zero pois, caso contrário, elas estariam representando camadas de enriquecimento. Nas *NAL Units* do tipo 20, por outro lado, pelo menos um dos ID's devem ter valor diferente de 0.

Dadas as condições de rede, são definidos os níveis de taxa de quadros, resolução e qualidade desejados pelo receptor. Tendo isto como base, se definem o PRID, DID, QID e TID objetivos. Tendo estes objetivos, cada camada é avaliada e deve ser excluída, como definido em (ITU-T, 2010), caso:

- O PRID da camada seja maior que o PRID alvo.
- O TID da camada seja maior que o TID alvo.
- O DID da camada seja maior que o DID alvo.
- O DID da camada seja igual ao DID alvo e o QID da camada seja maior que o QID alvo.

Satisfazendo o fato de que Prefix *NAL Units* e as correspondentes *NAL Units* associadas compõem um único elemento, a exclusão de uma *Prefix NAL Unit* pelas condições supracitadas acarreta a exclusão da *NAL Unit* associada.

3.3 Resumo

O vídeo H.264 é encapsulado em uma *Network Abstraction Layer* (NAL), que contém dados referentes aos quadros de vídeo, aos conjuntos de parâmetros e a dados suplementares. Quando trata de vídeo escalável, a NAL é responsável por identificar cada uma das camadas do vídeo H.264 SVC, permitindo que, no nível dos pacotes, seja possível extrair as camadas desnecessárias à exibição do vídeo com determinadas qualidade, resolução e taxa de quadros por segundo.

Com os conceitos abordados nos Capítulos 2 e 3, é possível apresentar, no Capítulo 4, os métodos de retransmissão de vídeo e como eles se aplicam no problema apresentado.

4 MODELOS DE RETRANSMISSÃO DE VÍDEO ADAPTÁVEL NA REDE

A retransmissão de vídeo é uma ferramenta utilizada para que dois terminais em uma rede consigam se comunicar através de vídeo sem que necessariamente um esteja ligado diretamente ao outro. Deste modo, um emissor pode enviar vídeo para o retransmissor e este o repassará para cada um dos receptores, sem que o emissor conheça a localização de cada um dos receptores.

Como o retransmissor se comunica com cada receptor individualmente, não há motivos para enviar vídeo em formato que sobrepuje os recursos do sistema. Para isso, são utilizados métodos de adaptação do vídeo às condições de rede.

Desse modo, podemos dividir o retransmissor de vídeo adaptável em duas partes: adaptação do vídeo às condições do vídeo e transmissão do vídeo entre o retransmissor e os terminais.

4.1 Transmissão de vídeo

A transmissão de vídeo, como qualquer formato de dado, para um conjunto de receptores é muito simples quando se utiliza transmissões *multicast* IP. Em uma rede que comporta o recurso da transmissão *multicast* IP, os roteadores são responsáveis por distribuir os pacotes de forma que todos os receptores associados a um grupo recebam o vídeo.

Mais especificamente, todos os terminais que desejam se comunicar se associam, inicialmente, a um grupo através de um protocolo que seja compreensível por roteadores e terminais, como o IGMP (Internet Group Management Protocol) (NEUMANN; ROCA, 2003). Após isso, os roteadores estabelecem tabelas de roteamento que evitam a redundância no envio de dados. Com isso cada roteador recebe os dados somente uma vez e os repassa adequadamente para os outros roteadores apenas uma vez. A transmissão de um emissor para todos os seus receptores no modelo *multicast* IP é exemplificada na Figura 4.1.

Grande parte da base instalada da Internet, atualmente, não comporta recursos como a transmissão no modelo *multicast*. É impossível garantir, portanto, que a subrede utilizada por uma aplicação comportará o recurso do *multicast*.

Para incorporar a arquitetura da Internet atual, é necessário se utilizar do modelo IP tradicional, para isso o emissor deve se conectar com cada um dos receptores individualmente, sem que os roteadores interfiram nos canais de comunicação. Com isso, são inseridos fluxos redundantes na rede, como mostrado no exemplo da Figura 4.2.

Uma forma de estabelecer funcionalidades semelhantes às do *multicast*, através do

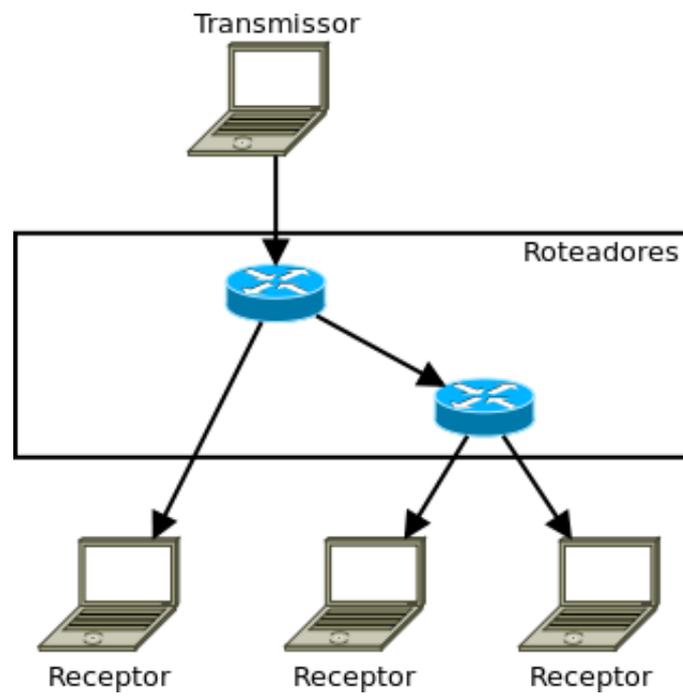


Figura 4.1: Transmissão em redes *multicast*

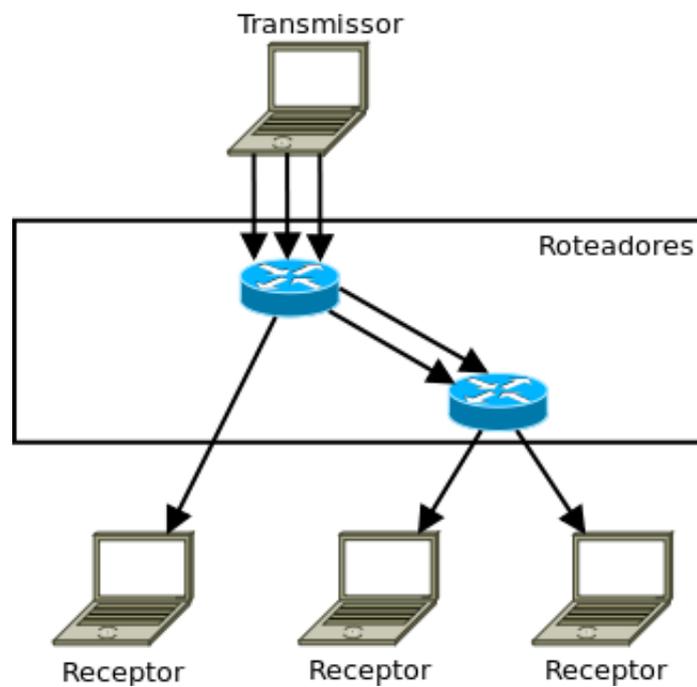


Figura 4.2: Transmissão em redes *unicast*

modelo *unicast*, ou seja, com redundância de dados, é inserir a figura do retransmissor. Neste caso, tanto o emissor como cada um dos receptores se associam ao retransmissor que fica responsável por receber vídeo do emissor e repassá-lo a tantos quantos forem os receptores. O modelo com um emissor e três receptores é apresentado na Figura 4.3.

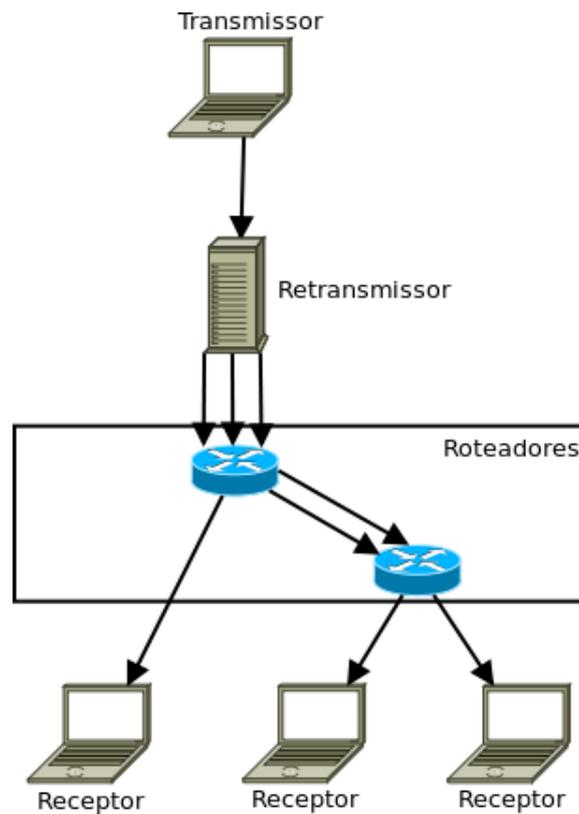


Figura 4.3: Retransmissor de vídeo

4.2 Adaptação do vídeo às condições da rede

Como o retransmissor provê um canal individual entre ele e cada um dos receptores e devido à heterogeneidade da infra-estrutura da Internet, a solução natural é adaptar cada um dos fluxos de vídeo às condições de rede apresentadas. Deste modo, um cliente que tem disponível taxa de transmissão maior receberá vídeo de melhor qualidade, enquanto o receptor cujo canal permite baixa qualidade de transmissão receberá vídeo com mais baixa taxa de bits. Para adaptação do vídeo às necessidades do cliente, podem-se destacar duas possibilidades: a transcodificação do vídeo e o roteamento do vídeo escalável.

4.2.1 Transcodificador de vídeo

Nos casos em que o vídeo não é codificado escalavelmente, a única possibilidade de adaptar o vídeo às condições de rede é transcodificar o vídeo.

A transcodificação de vídeo é o processo pelo qual um modelo de codificação de vídeo é transformado para outro modelo. Geralmente, este processo se dá pela decodificação do vídeo a um modelo intermediário (neste caso, a codificação YUV) e pela subsequente codificação para o modelo desejado. A transcodificação pode ocorrer dentro do mesmo

protocolo, desde que sejam aplicadas modificações nos parâmetros, como o fator de quantização, a resolução ou a taxa de quadros (VETRO; CHRISTOPOULOS; SUN, 2003).

O transcodificador recebe, portanto, o vídeo na maior taxa que a comunicação entre ele e o emissor permite. Por sua vez, ele decodifica o vídeo, adapta os parâmetros necessários, como taxa de quadros e resolução, e recodifica o vídeo sob os parâmetros que melhor se adaptam à taxa de transmissão entre ele e cada um dos receptores. O modelo do transcodificador de vídeo para adaptabilidade é apresentado no exemplo da figura 4.4, com três receptores em condições de rede e exibição distintas.

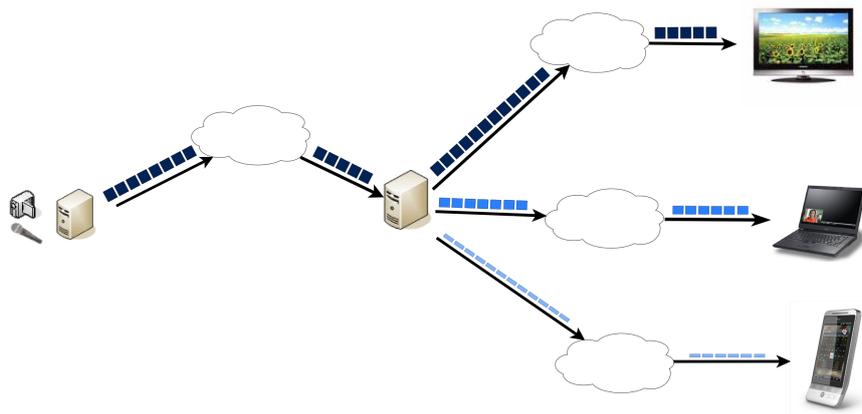


Figura 4.4: Transcodificador de vídeo

Um dos principais pontos positivos do transcodificador é possibilitar uma grande flexibilidade, pois está restrito somente pelos limites do codificador. Independentemente do vídeo recebido no retransmissor, é possível diminuir, na quantidade desejada, quaisquer dos parâmetros da codificação, como resolução, taxa de quadros ou parâmetro de quantização.

4.2.2 Roteador de vídeo escalável

Devido à natureza do vídeo sob codificação escalável e às abstrações de rede, como a NAL do H.264, que identifica as camadas, é possível, a partir de um *bitstream* de vídeo com duas ou mais camadas, extrair pacotes do vídeo de forma consistente para que o vídeo possa ser exibido com a qualidade que a taxa de transmissão permite.

Neste modelo, os terminais enviarão vídeo codificado de forma escalável para um roteador de vídeo, que lerá os dados recebidos e, avaliando as necessidades de cada receptor, identificará quais pacotes, devem ser repassados para cada um deles, seguindo o processo de identificação de camadas. Um exemplo com três receptores distintos é mostrado na figura 4.5.

O principal benefício deste modelo de retransmissor é o fato de não acrescentar grande complexidade computacional ao retransmissor.

4.3 H.323

O H.323 é uma das muitas recomendações da ITU-T voltadas para a transmissão áudio-visual. Este padrão prevê sessões de comunicação para áudio e vídeo sobre quaisquer redes de pacotes (THOM, 1996). O H.323, em verdade, encapsula uma série de

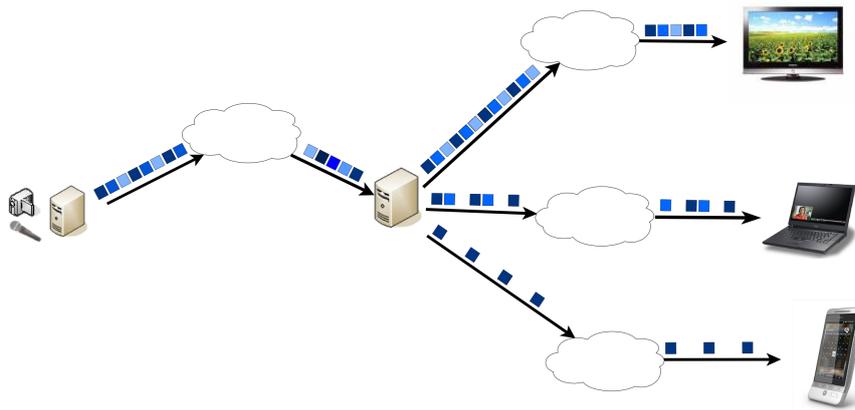


Figura 4.5: Roteador de vídeo escalável

outros protocolos (H.245, H.225, G.711, H.263, entre outros) para alcançar seu objetivo. Entre os papéis do H.323, podem-se destacar:

- Sinalização de chamadas.
- Sinalização de controle.
- Controle de multimídia.
- Transporte de multimídia.

Na recomendação são definidas quatro unidades básicas:

- **Terminal** - Emissores e receptores de áudio, vídeo e dados em tempo-real, podem-se comunicar sem a presença de nenhuma outra unidade.
- **Gatekeeper** - Provê serviços como: registro de terminais, resolução de endereços, limitação de largura de banda, entre outros. Em sistemas H.323 que são compostos por um *gatekeeper*, que não é obrigatório, os terminais devem pedir permissão do *gatekeeper* para fazer ou aceitar uma chamada, na negociação são definidas muitos parâmetros, como o limite de largura de banda dedicado a um terminal.
- **Gateway** - Provê interoperabilidade com terminais que utilizem outros protocolos ITU-T para comunicação, em especial:
- **MCU** - A *Multipoint Control Unit* faz o controle de multimídia entre os terminais e os *gateways*.

4.3.1 MCU

Existem dois tipos de conferências multipontos no H.323: as centralizadas e as descentralizadas. As conferências de vídeo centralizadas exigem a presença de um componente retransmissor de vídeo e podem funcionar em qualquer modelo de rede convencional. As descentralizadas, por sua vez, são aplicadas quando é possível que todos os terminais se comuniquem diretamente, como em uma rede *multicast*. Ainda há a possibilidade

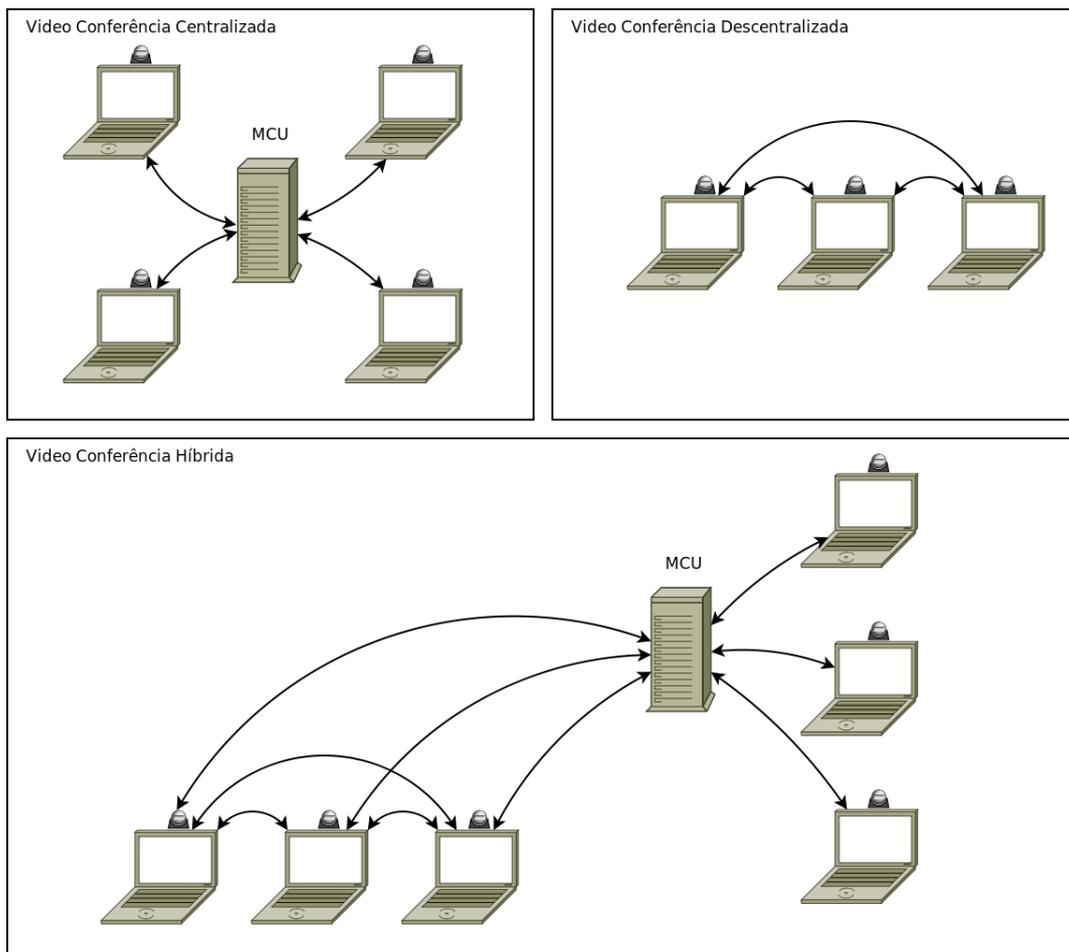


Figura 4.6: Formatos de vídeo conferência multi-pontos

de utilizar uma solução híbrida entre estes dois tipos. Os três formatos de conferência são mostrados na Figura 4.6 (THOM, 1996).

A *Multipoint Control Unit* (MCU) é a unidade de controle que permite que vários receptores e transmissores participem de uma mesma conferência de múltiplos pontos H.323 no formato centralizado, bem como, é a unidade que efetua o controle de ambos os tipos de conferência. Ela é composta de um MC (*Multipoint Controller*) e um conjunto, possivelmente vazio, de MP's (*Multipoint Processors*).

A MC é responsável pela administração da sessão em uma conferência H.323. Através da MC são passadas informações (codecs, taxas de bits, resoluções, entre outros) e são estabelecidos os canais de mídia e o modo comum de comunicação.

Já os MP's são responsáveis pelo mix de áudio e vídeo, chaveamento de vídeo e distribuição de dados para os terminais. Elas podem transformar o vídeo, de acordo com os receptores e de acordo com as necessidades de mixagem do vídeo. Devem, portanto, ser capazes de adaptar o vídeo a uma taxa de bits, resolução e taxa de quadros específicos, sendo, conseqüentemente, um modelo de retransmissor de vídeo adaptável de duas vias.

4.4 Resumo

Neste capítulo são apresentados os modelos de comunicação *unicast* e *multicast* e como os dois se aplicam às video conferências. Também é mostrado o fato de que a infraestrutura atual da Internet impede a utilização de *multicast* em larga escala, pela ausência de roteadores com suporte aos protocolos de *multicast*.

Como solução para o modelo *unicast*, se apresenta a ideia de um retransmissor de vídeo que centraliza os fluxos de vídeo, facilitando a video conferência. Como a retransmissão é feita individualmente para cada receptor, é possível adaptar os vídeos às necessidades de cada um. Para este fim, são propostos o transcodificador de vídeo, que decodifica o vídeo e o adapta às condições de rede de cada receptor, e o roteador de vídeo escalável, que utiliza as propriedades do vídeo escalável para, no nível dos pacotes, escolher as camadas que devem ser enviadas do vídeo.

Estes métodos podem ser utilizados como base para a adaptação de vídeo na MCU do H.323, um protocolo de sinalização incipiente para video conferências.

Com base nos dois modelos de retransmissão adaptativa apresentados, são propostos protótipos e testes que visam analisar e comparar ambos os métodos.

5 METODOLOGIA

No presente trabalho, o enfoque dado é o de comparar os métodos do Transcodificador de Vídeo e do Roteador de Vídeo Escalável, como soluções próprias para a retransmissão adaptável de vídeo em redes *unicast*. Antes de descrever a metodologia utilizada nos testes, contudo, é necessário descrever a metodologia utilizada para o desenvolvimento dos protótipos utilizados.

5.1 Modelos avaliados

Dois modelos de adaptação do vídeo aos requisitos dos receptores foram apresentados no capítulo 4: Transcodificador de Vídeo, que recodifica o vídeo da forma necessária, e o Roteador de Vídeo Escalável, que escolhe as camadas necessárias.

Para a melhor isenção do processo, a escolha natural foi utilizar o H.264 nos dois métodos, diferenciando-os apenas na utilização de SVC, no caso do Roteador de Vídeo Escalável. Vale ressaltar, que devido à sua complexidade computacional, o H.264 apresenta maiores usos de CPU nos processo de codificação e decodificação do que métodos de codificação mais antigos, como o MPEG-2 e o MPEG-4.

Devido à ausência de codificadores em tempo-real para o H.264/SVC com código-fonte disponível e/ou gratuitos, também não é possível levar em conta este parâmetro na comparação. Devido a este fato, o vídeo, nos dois casos, será pré-codificado e armazenado em um arquivo com a estrutura da NAL. Como o SVC é apenas uma extensão do H.264/AVC, a divisão do *bitstream* em *NAL Units* pode ser feita da mesma forma nos dois casos. Igualmente, a transformação do fluxo de *NAL Units* em um arquivo de saída, para posterior decodificação, pode ser feita num mesmo software nos dois casos.

Temos, então, dois softwares para o emissor e receptor de vídeo, idênticos para os dois modelos testados, provendo maior imparcialidade nos testes, sendo:

- **BitstreamSender** - O emissor de vídeo H.264. Lê um arquivo com o *bitstream* da NAL do H.264, o divide em *NAL Units* e envia cada uma delas em um pacote separado para o retransmissor de vídeo.
- **BitstreamReceiver** - Recebe, do transmissor, *NAL Units* no formato do H.264 em pacotes individuais e os une em um único arquivo.

Tendo excluído o codificador e o decodificador dos testes e utilizando o **BitstreamSender** e o **BitstreamReceiver**, a única diferença nos dois modelos fica, portanto, associada ao retransmissor.

- **Transcodificador de Vídeo** - O transcodificador recebe as *NAL Units* do H.264 separadas em pacotes e as passa ao decodificador. Para cada um dos formatos especificados por pelo menos um receptor, o vídeo será recodificado e enviado para todos os receptores que se adaptam a este formato.
- **Roteador de Vídeo Escalável** - O roteador receberá o vídeo dividido em *NAL Units* e, utilizando as condições citadas na seção 3.2, excluirá as camadas desnecessárias à exibição em um determinado receptor.

5.2 Testes propostos

O transcodificador e o roteador de vídeo escalável, diferem, devido às suas características, em:

- **Codificação e decodificação** - O roteador de vídeo escalável depende que o vídeo seja codificado com cabeçalhos e configuração do H.264/SVC. O transcodificador, ao contrário, depende apenas de configurações H.264/AVC.
- **Demanda de rede** - Por terem codificação diferente, a demanda de rede entre o Sender e o retransmissor e entre o retransmissor e o Receiver nos dois métodos será diferente.
- **Demanda computacional do retransmissor** - Por proporem métodos completamente diferentes, os dois retransmissores terão consumo de CPU e de memória bastante distintos.
- **Escalabilidade do retransmissor** - Como utilizam distintas metodologias, os dois retransmissores permitirão diferentes quantidades de receptores associados.
- **Atraso do Vídeo** - A diferença de tempo entre a codificação de um quadro no BitstreamSender e a sua exibição será distinta entre estes dois modelos, pois um processo distinto pode inserir maior atraso que o outro.

Como já explicado, não há codificadores e decodificadores de H.264/SVC em tempo-real com código-fonte livre e/ou gratuitos. Isto impede a avaliação concreta da diferença nos processos de codificação/decodificação e no atraso na exibição do vídeo, pois é impossível executar todo o sistema em tempo-real.

Será, para execução dos demais testes, utilizada a pré-codificação. Cada vídeo que se desejar utilizar para os testes será codificado antecipadamente. O resultado também será decodificado separadamente aos protótipos.

5.2.1 Definição dos vídeos utilizados

Um processo objetivo para testar a equidade de dois vídeos é comparar os valores de seus PSNR (*Peak Signal-to-Noise Ratio*). O PSNR é calculado com base no MSE (Mean Squared Error) do vídeo. Para a comparação entre os quadros x e y de um vídeo, devem ser calculados o MSE e o PSNR, mostrados nas equações abaixo (DARONCO, 2009):

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (5.1)$$

$$PSNR = 10 \log_{10} \frac{M^2}{MSE} \quad (5.2)$$

Nas equações 5.1 e 5.2, o valor N representa o número de pixels no vídeo, sendo que i é o índice que permite determinar o pixel tratado. M por sua vez indica o maior valor que a representação do pixel pode ter, no caso de um pixel com 8 bits este valor é 255.

Tendo o PSNR de cada quadro, o PSNR do vídeo é calculado como a média entre os PSNR's de cada quadro.

Embora este método de avaliação objetiva da qualidade apresente determinadas falhas (DARONCO, 2009), é suficientemente eficiente para que tenhamos valoração prática para uma comparação entre os vídeos.

Serão utilizados, para efeito de testes, vídeos com resolução de 720 pixels de largura por 480 pixels de altura, com taxa de 25 quadros por segundo. Em especial, foram escolhidos os vídeos:

- Ducks Take Off¹.
- Parkrun²
- Mobcal³

O primeiro vídeo apresenta bastante movimento, devido ao vôo dos patos. O segundo apresenta mudança brusca na característica de movimento do vídeo aos 5 segundos, quando o homem, que estava correndo, pára seu movimento. O terceiro vídeo apresenta pouco movimento mas, ainda assim, tem uma certa variação no vídeo.

Serão estabelecidos três níveis de adaptação em um vídeo, de forma que o receptor escolherá entre três aspectos possíveis. Com isso, o vídeo escalável será codificado com três camadas com os respectivos níveis e o vídeo não escalável será codificado no nível mais alto dos três, para poder ser adaptado no transcodificador.

Para cada um dos vídeos, serão utilizadas duas configurações de receptores. A primeira irá variar sobre a resolução da imagem, tendo como resoluções dos níveis:

1. 240x160
2. 480x320
3. 720x480

Para o caso da escalabilidade MGS, serão utilizados parâmetros de quantização correspondentes às necessidades de cada vídeo. Serão estabelecidos as taxas de bits desejadas para cada um dos níveis e o vídeo não-escalável será codificado com base neste valor, a qualidade alcançada será utilizada como base para a codificação idêntica do vídeo escalável. Estes valores poderão variar de acordo com o vídeo, como mostrado no capítulo 7.

Para igualar os vídeos transcodificados e os adaptados por exclusão de camadas, será utilizada a medida PSNR discutida anteriormente, o parâmetro de quantização nos dois casos será modificado de forma a igualar os PSNR's dos vídeos.

¹Disponível em http://media.xiph.org/video/derf/y4m/720p/ducks_take_off_420_720p.y4m, no dia 29 de novembro de 2010

²Disponível em http://media.xiph.org/video/derf/y4m/720p/720p50_parkrun_ter.y4m, no dia 29 de novembro de 2010

³Disponível em http://media.xiph.org/video/derf/y4m/720p/720p50_mobcal_ter.y4m, no dia 29 de novembro de 2010

5.2.2 Demanda de rede

Um dos pontos em que há divergências entre os dois modelos de retransmissor é o da demanda de rede necessária para que o vídeo seja transmitido do BitstreamSender para o retransmissor e, por sua vez, deste para o BitstreamReceiver.

A partir das três configurações de vídeo, definidas na seção 5.2.1, com o receptor recebendo, em três testes distintos, vídeo com cada uma das especificações de qualidade, é feita a análise da banda de rede utilizada pela ferramenta WireShark⁴, que indica o tráfego de rede de uma determinada porta.

Medindo os tráfegos, será possível aferir, com precisão, as bandas de rede utilizadas a cada segundo da transmissão nos casos do roteador de vídeo escalável e do transcodificador de vídeo. A partir destes dados se estabelecerão os gráficos de bits por segundo de tráfego na rede para cada configuração em cada um dos modelos estabelecidos.

5.2.3 Demanda computacional do retransmissor

Outra característica passível de análise e comparação é a demanda computacional necessária para que a retransmissão seja executada com o vídeo adaptado sendo enviado para os BitstreamReceiver's com as configurações supracitadas na seção 5.2.1.

Para cada conjunto de configurações, com três níveis de qualidade ou resolução, serão efetuados três testes, um para cada nível. Nos três testes o retransmissor receberá o vídeo do BitstreamSender, o adaptará e enviará para o receptor.

Serão avaliados, com base nos dados do Gnome System Monitor⁵, cada uma das configurações para os dois modelos de retransmissor. Com isso, será possível estabelecer gráficos de porcentagem de uso da CPU do sistema em cada um dos processos.

5.2.4 Escalabilidade do processo

Como os dois retransmissores utilizam porcentagem de CPU e banda de rede distintas, o número possível de receptores ligados a um mesmo retransmissor é, também, um valor interessante para análise e comparação.

Utilizando as mesmas configurações definidas na seção 5.2.1, serão inseridos, de três em três BitstreamReceiver's (um para cada um dos níveis), até que não mais se possam acrescentar novos receptores sem que o computador colapse.

Com este processo, será possível estabelecer uma tabela de quantidade de receptores aceitável para cada um dos modelos de retransmissão.

⁴Disponível em <http://www.wireshark.org/>, no dia 29 de novembro de 2010

⁵Disponível em <http://library.gnome.org/users/gnome-system-monitor/> no dia 29 de novembro de 2010

6 DESENVOLVIMENTO

Para o teste adequado do sistema, é necessário o desenvolvimento de quatro ferramentas para os protótipos:

- BitstreamSender
- BitstreamReceiver
- BitstreamScalableRouter
- BitstreamTranscoder

O primeiro passo para o desenvolvimento dos protótipos foi uma análise de bibliotecas e ferramentas conhecidas que auxiliam com a codificação e decodificação de vídeo não-escalável, com a extração de camadas em vídeo escalável e com o transporte dos pacotes de vídeo. Com a decisão de utilizar partes da JSVM e as bibliotecas-base do sistema IVA (Sistema Interativo de Vídeo e Áudio) (ROESLER; HUSEMANN; COSTA, 2009).

Em um segundo momento, utilizando as bibliotecas do IVA e modificando parte do código do BitstreamExtractor para o roteador de vídeo escalável, foram desenvolvidos os dois protótipos para teste.

6.1 Bibliotecas utilizadas

O reuso é uma das ferramentas mais poderosas da engenharia de software, pois evita o retrabalho e aumenta a eficácia dos códigos. Desta forma, aproveitando o máximo de códigos prontos disponíveis, foram desenvolvidos os dois protótipos do IVA. Os códigos reaproveitados no processo de desenvolvimento foram:

- Bibliotecas do Projeto IVA, que incluem bibliotecas de codificação e decodificação não escaláveis de vídeo, baseadas na FFMPEG.
- Partes do código do BitstreamExtractor da JSVM, aplicativo citado na Seção 2.3.

6.1.1 Bibliotecas do projeto IVA

O IVA é um projeto de ensino a distância em áudio e vídeo que comporta uma série de dispositivos e modelos de rede (ROESLER; HUSEMANN; COSTA, 2009), desenvolvido pelo grupo PRAV (Projetos em Áudio e Vídeo) da Universidade Federal do Rio Grande do Sul (UFRGS).

Por ser um projeto de áudio e vídeo, o IVA conta com diversas bibliotecas já desenvolvidas com propósitos de captura de vídeo, codificação e decodificação de vídeo não-escalável (utilizando FFMPEG) e transporte de dados de vídeo pela rede.

As bibliotecas necessárias à implementação dos protótipos para testes são:

- **common** - A biblioteca common é responsável por funcionalidades necessárias em diversas bibliotecas, ou seja, bibliotecas de uso comum. Entre as funcionalidades da common, podem-se citar: tratamento de erros, logs do sistema, tratamento de strings, tratamento de arquivos, entre outros.
- **thread** - Biblioteca que encapsula, com orientação a objeto, as funcionalidades da biblioteca PThread (LEWIS; BERG, 1998), além de envolver funcionalidades para temporização. Esta biblioteca contém as classes Thread, Mutex, Interval, entre outros, necessárias à execução simultânea de codificadores e analisadores de dados nos dois modelos de retransmissão implementados.
- **queue** - Esta biblioteca define as filas de áudio e vídeo, uma das unidades mais importantes do sistema IVA. A biblioteca incorpora um sistema de gerenciamento de memória interno que reserva um grande espaço de memória e evita chaveamento desnecessário ao núcleo do sistema operacional. Além disso, a queue é implementada com controle de acesso para vários leitores, utilizando ferramentas de sincronismo. A mais importante estrutura desta biblioteca é a queue_t.
- **net** - Biblioteca com funcionalidades de transmissão e recepção em redes *unicast* e *multicast*, sobre protocolo UDP. Provê mecanismos para ordenação dos pacotes e checagem da integridade dos dados. Conta com dois mecanismos principais de acesso à rede: a NetSend, responsável por separar os quadros de vídeo e áudio, provenientes de uma fila, em pacotes e os enviar pela rede; e a NetRecv, responsável por receber os pacotes de dados da rede e uní-los, inserindo-os em uma queue de saída.
- **Encode** - Responsável por codificar vídeo e áudio. Para o protótipo do transcoder, a classe mais importante da biblioteca é a EncodeVideo, que recebe uma queue_t com quadros do vídeo em formato YUV:4:2:0 (TUDOR, 1995) e o codifica para o formato especificado, no caso H.264, inserindo o *bitstream* de saída em uma outra queue_t.
- **Decode** - É a biblioteca responsável pela decodificação de vídeo. Realiza o processo inverso da biblioteca Encode, ou seja, há uma classe DecodeVideo que recebe uma queue_t com os dados de vídeo codificados em H.264 e os decodifica, produzindo uma nova queue_t com os quadros do vídeo em formato YUV:4:2:0.

6.1.2 Código do BitstreamExtractor

O BitstreamExtractor do JSVM é responsável por receber vídeo em formato YUV de um arquivo, separá-lo em *NAL Units* e extrair as camadas desejadas do vídeo, de acordo com os parâmetros da linha de comando.

Do ponto de vista do desenvolvedor, existe um objeto chamado Extractor no código deste aplicativo, que recebe os parâmetros e organiza as chamadas de funções internamente para a seleção das camadas corretas.

O programa então tem três fases: a de pré-análise, a análise e a extração das camadas. O fato de o arquivo de vídeo passar por três leituras, torna a utilização deste mesmo modelo impraticável para implementação em redes. É possível, no entanto, aproveitar os trechos do Extractor que dividem o *bitstream* do arquivo em *NAL Units* e os que fazem a seleção das camadas aproveitadas e não aproveitadas, dados os ID's alvos das camadas.

6.1.2.1 Divisão do *bitstream* em *NAL Units*

Para retirar os dados do *bitstream* e separar ele em *NAL Units*, o Extractor utiliza a *ReadBitstreamIf*, uma classe base com métodos que retiram, de uma *bitstream*, as *NAL Units* já separadas. No software *BitstreamExtractor* convencional, a classe que herda as propriedades do *ReadBitstreamIf* é a *ReadBitstreamFile*, que lê os dados do arquivo e os separa.

Utilizando uma *queue_t* e inserindo nela os pacotes retirados da rede através da *NetRecv* do IVA, é possível extrair pacote a pacote da rede, cada um contendo uma *NAL Unit* já separada como necessário para o restante do processo do Extractor. Herda-se, portanto, a classe *ReadBitstreamIf* e são utilizadas a *NetSend* e a *queue_t*, para receber os pacotes de vídeo separados por *NAL Unit* através da rede.

6.1.2.2 Identificação e exclusão de camadas

Na classe *Extractor*, o método que examina as *NAL Units* é o *xExtractLayerLevel*, ele primeiro recebe as *NAL Units* de um objeto *ReadBitstreamIf*, neste caso herdados por um *ReadBitstreamNetwork*. Pelo tipo da *NAL Unit*, identifica como deve proceder e, caso seja uma *NAL Unit* com dados do SVC, identifica se a camada deve ser excluída, com base nos parâmetros de QID, DID e TID alvos. Além disso, este método associa *NAL Units* que devem ser tratadas como tal, como o caso da *Prefix NAL Unit* e os correspondentes *slices* de camadas-base.

Após a leitura de cada *NAL Unit* ou dupla de *NAL Units* associadas, elas estão marcadas para exclusão ou para manutenção no fluxo, portanto, somente as marcadas para manutenção são inseridas na classe base *WriteBitstreamIf*, neste caso herdada por uma *WriteBitstreamToNetwork*, através do método *WritePacket*.

6.1.2.3 Junção das *NAL Units* em um *bitstream* de saída

Igualmente ao caso do *ReadBitstreamIf* existe uma classe base chamada *WriteBitstreamIf*, que é utilizada para a junção de *NAL Units* através de um *bitstream*.

Portanto, é possível, também, definir um objeto *WriteBitstreamToNetwork* que utiliza as *queue_t*'s e a *NetSend* do IVA para enviar cada uma das *NAL Units* pela rede, apenas herdando dados do *WriteBitstreamIf*.

6.2 Desenvolvimento e implementação dos protótipos

Tendo como base as bibliotecas selecionadas e sabendo que os dois modelos utilizam o mesmo protocolo, a *Network Abstraction Layer*, foram desenvolvidos quatro aplicativos. Os dois primeiros fazem o envio e o recebimento de *NAL Units* do vídeo H.264, tanto SVC quanto AVC. Outro é o correspondente ao Transcodificador de Vídeo, chamado de *BitstreamTranscoder*. O último é o *BitstreamScalableRouter*, que implementa o Roteador de Vídeo Escalável.

6.2.1 BitstreamSender

O `BitstreamSender` é responsável por ler as *NAL Units* individualmente do arquivo e as enviar pela rede de forma que no lado do receptor elas sejam recebidas inteiras.

Para isso, será executado um processo temporizado em que cada quadro lido das *NAL Units* será enviado, de acordo com a taxa de quadros estipulada, para o retransmissor.

Para ler cada uma das *NAL Units*, é utilizada a classe `ReadBitstreamFile`. A partir desta classe são identificados o tipos das *NAL Units* e, a partir disto, é possível estabelecer o tempo entre o envio de *NAL Units* distintas, para que a taxa de quadros especificado para a transmissão seja alcançado.

Para o envio das *NAL Units* pela rede, é possível utilizar a classe `WriteBitstreamToNetwork`, explicada na seção 6.1.2.3.

6.2.2 BitstreamReceiver

O `BitstreamReceiver` é responsável por receber *NAL Units* da rede e as unir, ordenadamente, em um arquivo H.264. Para isso, será executado um processo para receber os quadros do vídeo conforme chegam, utilizando a classe `ReadBitstreamNetwork`.

Depois de recebidas cada uma das *NAL Units*, com base no fato de que a `ReadBitstreamNetwork` (devido à utilização da biblioteca Net do IVA) já tem seus dados ordenados, basta armazená-las no arquivo de saída utilizando a classe `WriteBitstreamToFile`.

6.2.3 BitstreamTranscoder

O `BitstreamTranscoder` é responsável por receber vídeo em H.264/AVC com as *NAL Units* separadas e ordenadas de acordo com a sequência do arquivo lido no `BitstreamSender`.

Com as *NAL Units* recebidas, é possível decodificar o vídeo e inserí-lo em uma fila intermediária. Como a biblioteca `queue` do IVA comporta que vários leitores acessem uma mesma `queue_t` de forma síncrona e recebendo os mesmos quadros, é possível que vários codificadores se associem à fila intermediária e codifiquem, independentemente, o vídeo, sob os parâmetros adequados, ao receptor correspondente.

Muitos receptores terão os mesmos parâmetros de configuração nos testes, entretanto, no mundo real, o transcodificador não tem um conjunto de configurações pré-estabelecidas e portanto há um sem-número de configurações possíveis, sendo difícil ter receptores com parâmetros repetidos. Esta característica, portanto, não foi implementada no `BitstreamTranscoder`. No entanto, nos casos em que o *bitstream* recebido é suficiente para as necessidades de um ou mais receptores, os receptores receberão o vídeo sem passar pelos processos de decodificação e decodificação.

6.2.4 BitstreamScalableRouter

O `BitstreamScalableRouter` é responsável por receber vídeo em H.264/SVC com as *NAL Units* separadas e ordenadas de acordo com a sequência do arquivo lido no `BitstreamSender`.

Utilizando a própria classe `Extractor`, igual à do `BitstreamExtractor`, e utilizando apenas o componente para identificação de camadas, descrito na Seção 6.1.2.2, o que evita que sejam feitas três leituras de todo o vídeo, é possível receber e enviar vídeo pela rede com os pacotes de vídeo desnecessários excluídos.

Para que todos os receptores sejam atendidos, com seus vídeos adaptados, é necessário que existam vários objetos da classe `Extractor`, no entanto, todos os objetos `ReadBitstre-`

amNetwork, referenciados dentro destes objetos devem acessar uma mesma fila, sendo somente consumidores distintos desta.

7 RESULTADOS

Os vídeos Ducks Take Off, Parkrun e Mobcal serão avaliados sobre três aspectos definidos anteriormente: a Demanda de Rede para transmissão do vídeo (Seção 7.2), a Demanda Computacional do retransmissor (Seção 7.3) e a Escalabilidade do processo (Seção 7.4).

7.1 Codificação dos vídeos

O primeiro passo para a execução dos testes é a codificação dos vídeos nos dois formatos: H.264/AVC e SVC. A definição de parâmetros que tornem os testes igualitários é um fator importante para a correta execução dos testes. São utilizados, para isso, quatro parâmetros que definem cada um dos níveis de codificação dos vídeos supra-citados: Largura e altura dos quadros, taxa de quadros do vídeo e o PSNR do vídeo codificado com relação ao vídeo original. Estas configurações de vídeo são apresentadas nas Tabelas 7.1, 7.2, 7.3, 7.4, 7.5 e 7.6.

Tabela 7.1: Definição dos níveis para o vídeo Ducks Take Off, com resolução variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	240	480	720
Altura (em pixels)	160	320	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	18,7	22,8	31,3

Tabela 7.2: Definição dos níveis para o vídeo Ducks Take Off, com qualidade variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	720	720	720
Altura (em pixels)	480	480	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	22,3	26,7	31,1

Para o funcionamento do transcodificador de vídeo, o vídeo utilizado pelo transmissor deve estar codificado com o maior dos níveis, no caso o nível 3. Já no caso do roteador de vídeo escalável, o vídeo deve conter três camadas que, juntas produzem os três níveis

Tabela 7.3: Definição dos níveis para o vídeo Parkrun, com resolução variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	240	480	720
Altura (em pixels)	160	320	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	19,8	21,4	34,3

Tabela 7.4: Definição dos níveis para o vídeo Parkrun, com qualidade variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	720	720	720
Altura (em pixels)	480	480	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	24,0	29,3	34,5

Tabela 7.5: Definição dos níveis para o vídeo Mobcal, com resolução variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	240	480	720
Altura (em pixels)	160	320	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	17,0	24,1	30,8

Tabela 7.6: Definição dos níveis para o vídeo Mobcal, com qualidade variando

	Nível 1	Nível 2	Nível 3
Largura (em pixels)	720	720	720
Altura (em pixels)	480	480	480
Taxa de quadros (em quadros/segundo)	25	25	25
PSNR (em decibéis)	21,5	26,9	32,0

correspondentes. Foi possível enxergar a grande diferença entre o codificador H.264/AVC da FFMPEG e o codificador de vídeo escalável da JSVM. A seguir são apresentados os tempos de codificação em cada um dos modelos para cada um dos vídeos:

- **Vídeo Ducks Take Off, com resolução variando.**
 - Tempo de codificação do vídeo não-escalável: 6,142 segundos.
 - Tempo de codificação do vídeo escalável: 21 minutos e 6 segundos.
- **Vídeo Ducks Take Off, com qualidade variando.**
 - Tempo de codificação do vídeo não-escalável: 6,796 segundos.

- Tempo de codificação do vídeo escalável: 34 minutos e 16 segundos.
- **Vídeo Parkrun, com resolução variando.**
 - Tempo de codificação do vídeo não-escalável: 7,541 segundos.
 - Tempo de codificação do vídeo escalável: 23 minutos e 41 segundos.
- **Vídeo Parkrun, com qualidade variando.**
 - Tempo de codificação do vídeo não-escalável: 7,478 segundos.
 - Tempo de codificação do vídeo escalável: 37 minutos e 19 segundos.
- **Vídeo Mobcal, com resolução variando.**
 - Tempo de codificação do vídeo não-escalável: 6,986 segundos.
 - Tempo de codificação do vídeo escalável: 18 minutos e 12 segundos.
- **Vídeo Mobcal, com qualidade variando.**
 - Tempo de codificação do vídeo não-escalável: 8,141 segundos.
 - Tempo de codificação do vídeo escalável: 25 minutos 54 segundos.

7.2 Demanda de rede

7.2.1 Ambiente de testes

Foram utilizados três computadores em cada um dos modelos de retransmissor de vídeo, todos eles ligados a um mesmo switch de rede, como apresentado na figura 7.1.

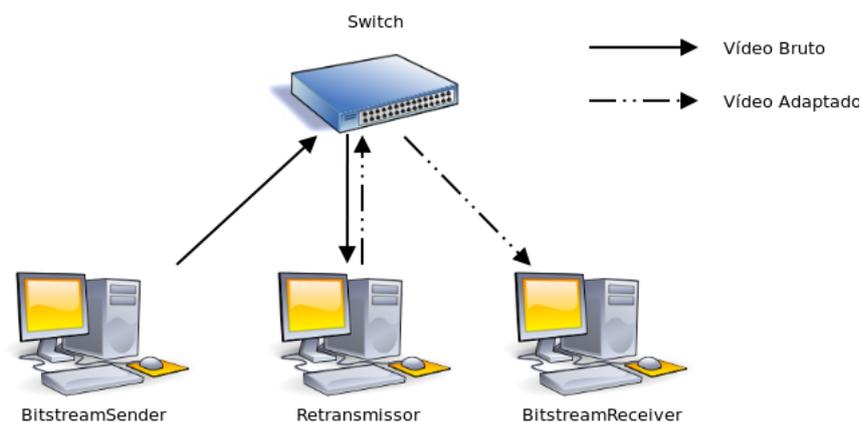


Figura 7.1: Ambiente para os testes de demanda de rede

O switch utilizado nos testes foi o Switch Dell PowerConnect 2824, já a configuração de cada uma das máquinas é apresentada na Tabela 7.7.

Com os vídeos pré-codificados da Seção 7.1, é possível iniciar os testes. Utilizando a ferramenta de gráficos de Entrada/Saída do software Wireshark, executado na máquina do BitstreamReceiver, foi possível estabelecer a demanda de rede necessária para a transmissão de vídeo, para ambos os retransmissores adaptáveis.

Tabela 7.7: Configurações das máquinas de teste

Software	Configuração da Máquina
BitstreamSender	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.04 (Lucid Lynx) 32-bit
Retransmissor	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.10 (Maverick Meerkat) 32-bit
BitstreamReceiver	Notebook Sony Vaio VGN-FE31B Intel Core 2 Duo T5500 1.66GHz 1,47 Gb de Memória RAM Placa de vídeo NVidia GeForce Go 7400 Ubuntu Linux 10.04 (Lucid Lynx) 64-bit

7.2.2 Resultados

Nas Figuras 7.2 e 7.3 são exibidos os gráficos que comparam, durante o período de transmissão do vídeo Ducks Take Off (em torno de 10 segundos), a taxa de bits ocupados da rede em cada um dos dois processos. Nestes gráficos são analisados, separadamente, o consumo de banda de saída do retransmissor para receptores com cada um dos níveis estipulados nas Tabelas da Seção 7.1. Os gráficos apresentam a taxa de bits enviada a cada segundo da transmissão por ambos os retransmissores, sendo mostrada, ao lado, a taxa de bits total das transmissões, em bps.

Os gráficos completos para os vídeos Parkrun e Mobcal se encontram no Apêndice A e apresentam resultados muito semelhantes aos do vídeo Ducks Take Off, com relação a diferença entre os dois métodos.

Por se tratar de uma transmissão VBR (*Variable Bitrate*), em que a taxa de bits da transmissão se adapta às condições e formas do vídeo, os gráficos apresentam picos e baixas que representam o movimento apresentado. A parada do corredor no vídeo Parkrun, por exemplo, torna a codificação muito simplificada e isto é representado por uma baixa nos valores da taxa de bits nos últimos 5 segundos de vídeo. Da mesma forma, o movimento no vídeo Ducks Take Off é representado por picos do gráfico correspondente. O vídeo Mobcal, que apresenta estrutura linear, quanto aos movimentos, é representado com menor variação nos gráficos.

Observa-se claramente, que no nível 1, há pouca diferença entre o roteador de vídeo escalável e o transcodificador de vídeo, isto se deve ao fato de que a camada-base do vídeo escalável é codificada igualmente aos quadros de vídeos não escaláveis.

Nos demais níveis se nota, entretanto, em todos os casos, uma diferença acentuada no consumo de rede, variando na faixa de 10% a 30%. Na tabela 7.8 são apresentadas as diferenças percentuais de consumo de banda do transcodificador de vídeo para o roteador de vídeo escalável.

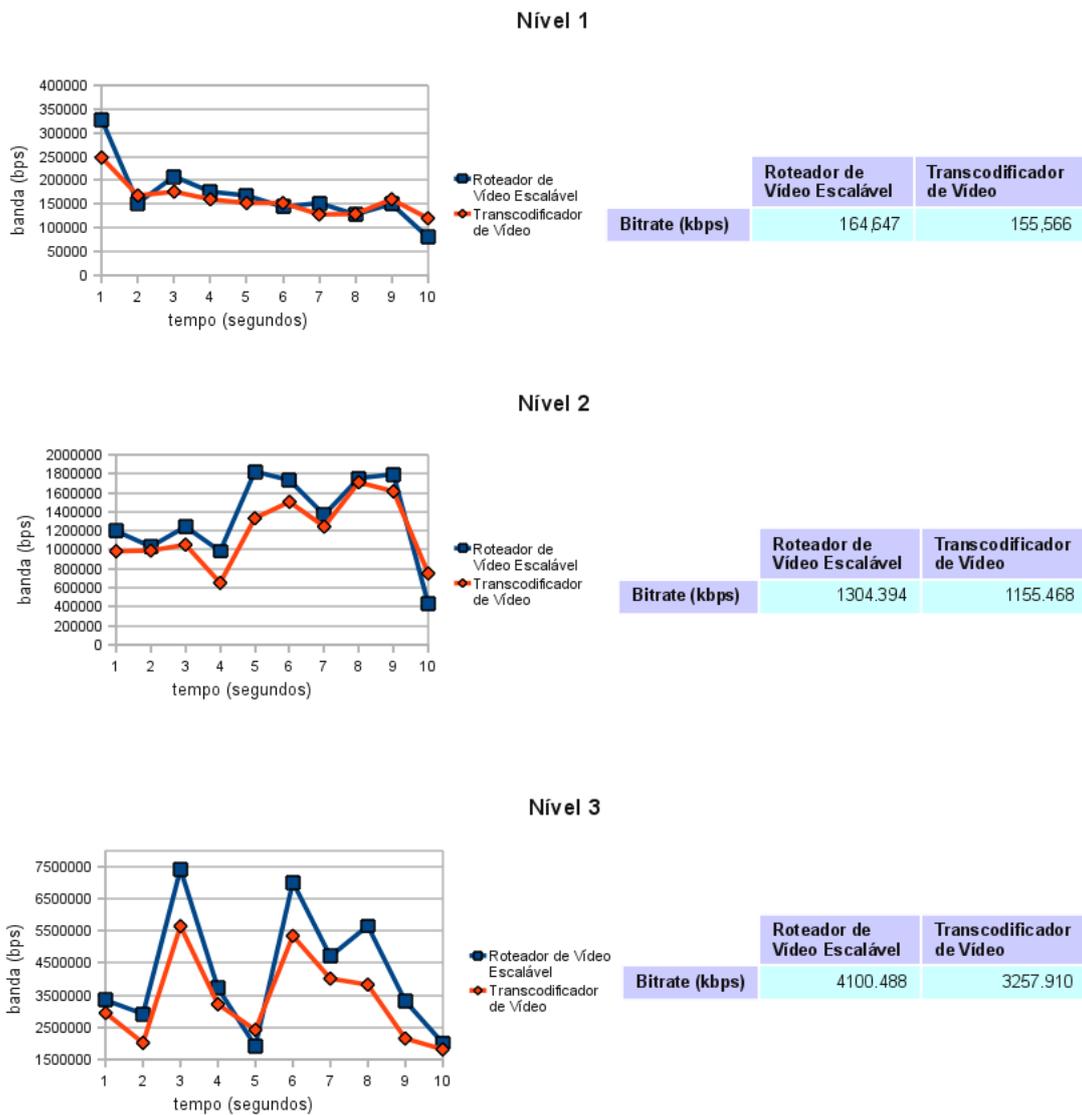
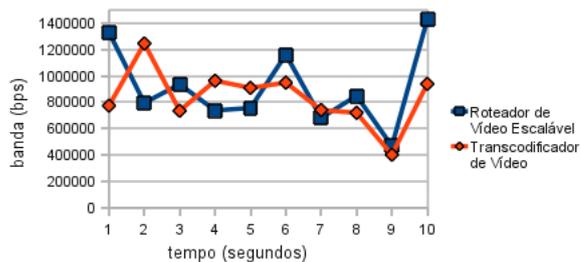


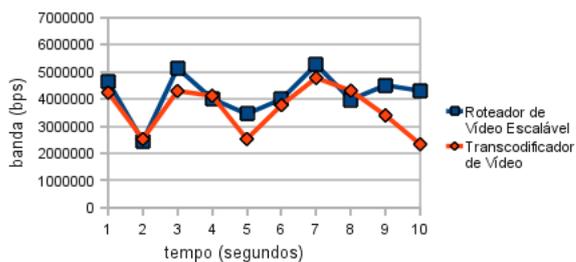
Figura 7.2: Demanda de rede nos três níveis do vídeo Ducks Take Off com resolução variando

Nível 1



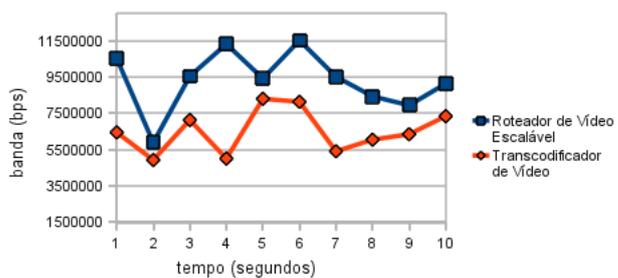
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	279.687	272.656

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	4078.906	3548.730

Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	9102.050	6359.570

Figura 7.3: Demanda de rede nos três níveis do vídeo Ducks Take Off com qualidade variando

Tabela 7.8: Acréscimo de taxa de bits inserido pelo Vídeo Escalável

Vídeo	Nível 1	Nível 2	Nível 3
Ducks Take Off (Resolução)	5.83%	12.88%	25.82%
Ducks Take Off (Qualidade)	1.57%	8.55%	20.23%
Parkrun (Resolução)	5.24%	16.35%	31.15%
Parkrun (Qualidade)	3.35%	19.09%	29.07%
Mobcal (Resolução)	3.50%	14.36%	26.69%
Mobcal (Qualidade)	3.16%	11.82%	33.39%

Estes resultados foram obtidos após 5 execuções de cada teste, que não apresentaram diferença, por se tratar do mesmo fluxo de vídeo.

7.3 Demanda computacional

7.3.1 Ambiente de testes

Foram utilizados três computadores em cada um dos modelos de retransmissor de vídeo, todos eles ligados a um mesmo switch de rede, como apresentado na figura 7.4.

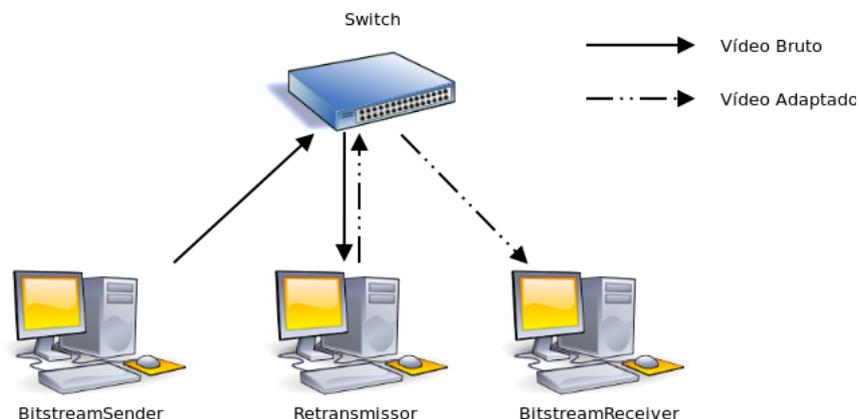


Figura 7.4: Ambiente para os testes de demanda de rede

O switch utilizado nos testes foi o Switch Dell PowerConnect 2824, já a configuração de cada uma das máquinas é apresentada na Tabela 7.9.

Utilizando, também, os vídeos pré-codificados, é possível estabelecer o consumo de CPU necessário para a utilização de ambos os métodos. Com as ferramentas do Gnome System Monitor, foram estabelecidos os gráficos de consumo de CPU pelo tempo.

7.3.2 Resultados

Nas Figuras 7.5 e 7.6 são exibidos os gráficos que comparam, durante o período de transmissão do vídeo (em torno de 10 segundos), a porcentagem de utilização das CPU's para cada um dos dois processos. Nos gráficos são analisados, separadamente, a porcentagem de processamento utilizada pelo retransmissor para se comunicar com receptores com cada um dos níveis estipulados nas Tabelas da Seção 7.1.

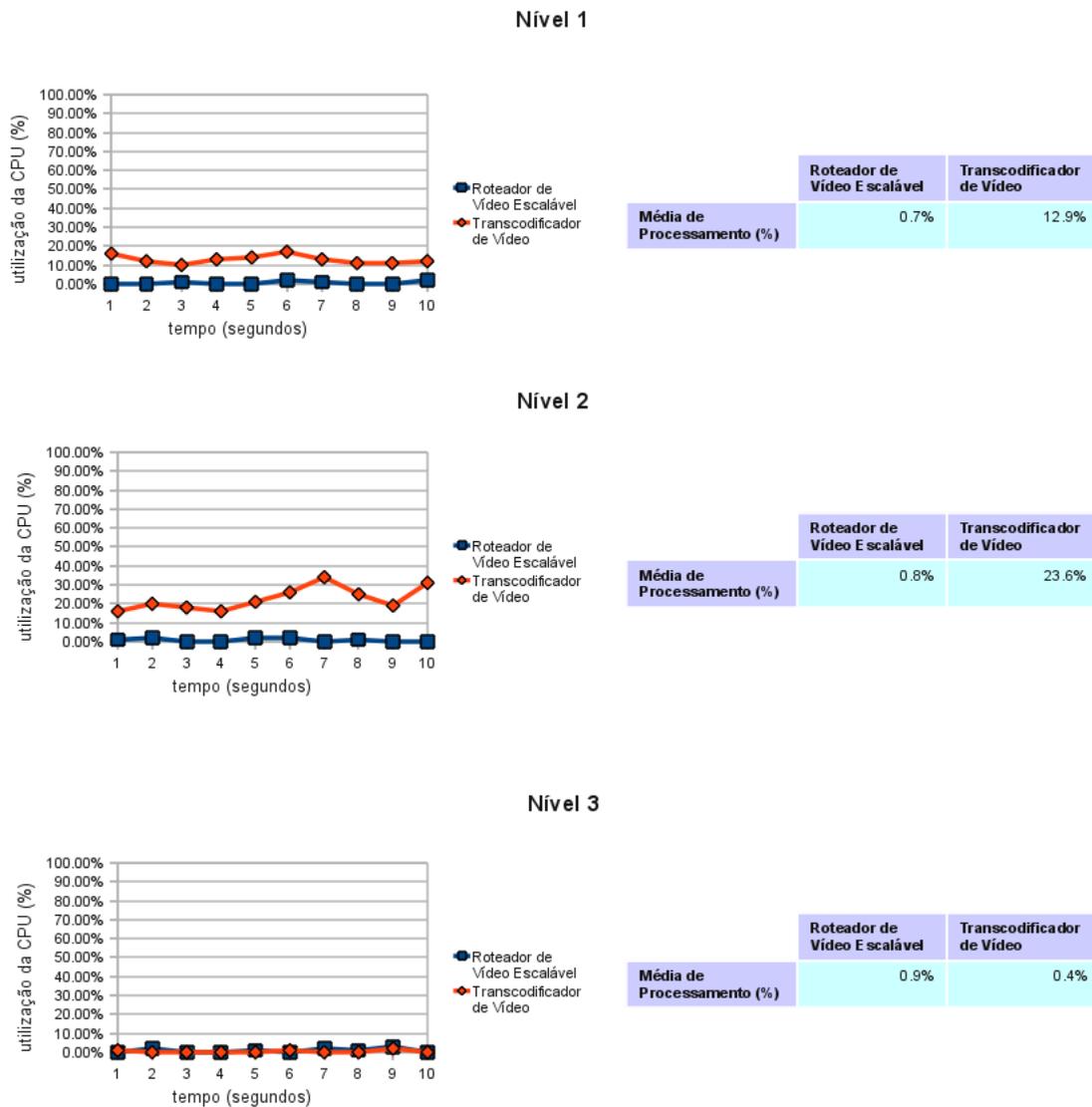
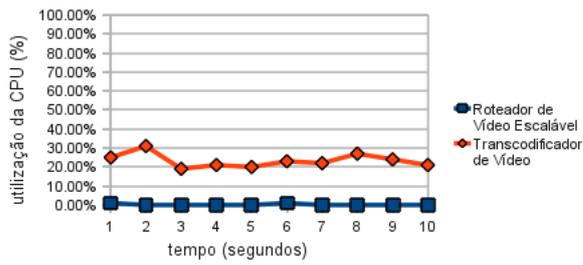


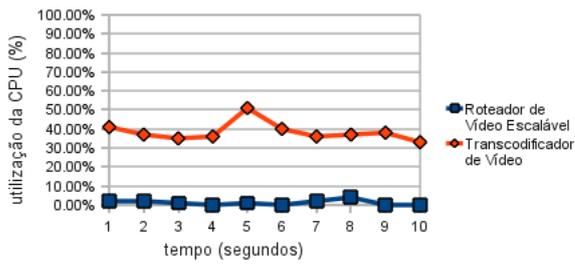
Figura 7.5: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Ducks Take Off com resolução variando

Nível 1



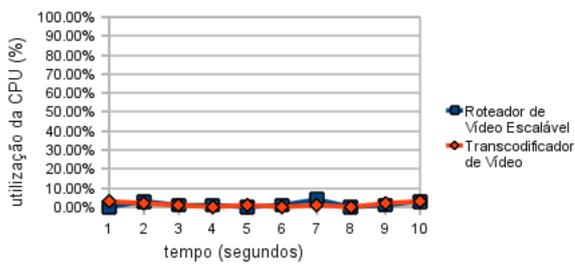
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	0.2%	23.3%

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1.2%	38,4%

Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1.5%	1.3%

Figura 7.6: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Ducks Take Off com qualidade variando

Tabela 7.9: Configurações das máquinas de teste

Software	Configuração da Máquina
BitstreamSender	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.04 (Lucid Lynx) 32-bit
Retransmissor	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.10 (Maverick Meerkat) 32-bit
BitstreamReceiver	Notebook Sony Vaio VGN-FE31B Intel Core 2 Duo T5500 1.66GHz 1,47 Gb de Memória RAM Placa de vídeo NVidia GeForce Go 7400 Ubuntu Linux 10.04 (Lucid Lynx) 64-bit

Nota-se, claramente, uma diferença substancial na utilização do processador feita pelos dois métodos nos níveis 1 e 2. Enquanto o transcodificador de vídeo utiliza vastamente as unidades de processamento, o roteador de vídeo escalável praticamente não faz uso destas. Para o nível 3, há pouca ou nenhuma diferença na média, pois o transcodificador mantém o vídeo com as configurações recebidas, apenas repassando os pacotes para o BitstreamReceiver.

Os dados de consumo da CPU são representativos de um vídeo codificado em H.264, o que explica o alto consumo do processador durante a execução do transcodificador para os níveis 2 e 3, que não se repete em codificadores como o MPEG-4 ou MPEG-2.

Os resultados correspondentes aos demais vídeos podem ser encontrados no Apêndice B.

7.4 Escalabilidade do processo

Outro parâmetro a se avaliar é o número de receptores que cada um dos retransmissores suporta. Foram feitas tentativas sucessivas de executar o retransmissor, aumentando em três o número de receptores (cada um recebendo um dos níveis de vídeo citados na Seção 7.1) em cada execução. O acréscimo de receptores de três em três, com cada um dos níveis, é uma forma de comparar os dois métodos de forma igualitária, evitando comparar receptores com níveis distintos.

O parâmetro limitante, para que a execução seja considerada falha, é a execução satisfazer requisitos de tempo-real e qualidade. Se o vídeo chega no receptor em tempo adequado (a transmissão dura os mesmos 10 segundos da exibição do vídeo) e não há perdas de quadros ou camadas do vídeo, a execução é considerada bem sucedida e é executada uma nova iteração. Quando estas condições não são satisfeitas, a execução é considerada falha e considera-se que, a partir deste número de receptores, o processo não

escala mais.

7.4.1 Ambiente de Testes

Foram utilizados cinco computadores em cada um dos modelos de retransmissor de vídeo, todos eles ligados a um mesmo switch de rede, como apresentado na figura 7.7. Devido à indisponibilidade de máquinas, cada um dos BitstreamReceiver's foi encarregado de receber todos os streams de um mesmo nível do vídeo.

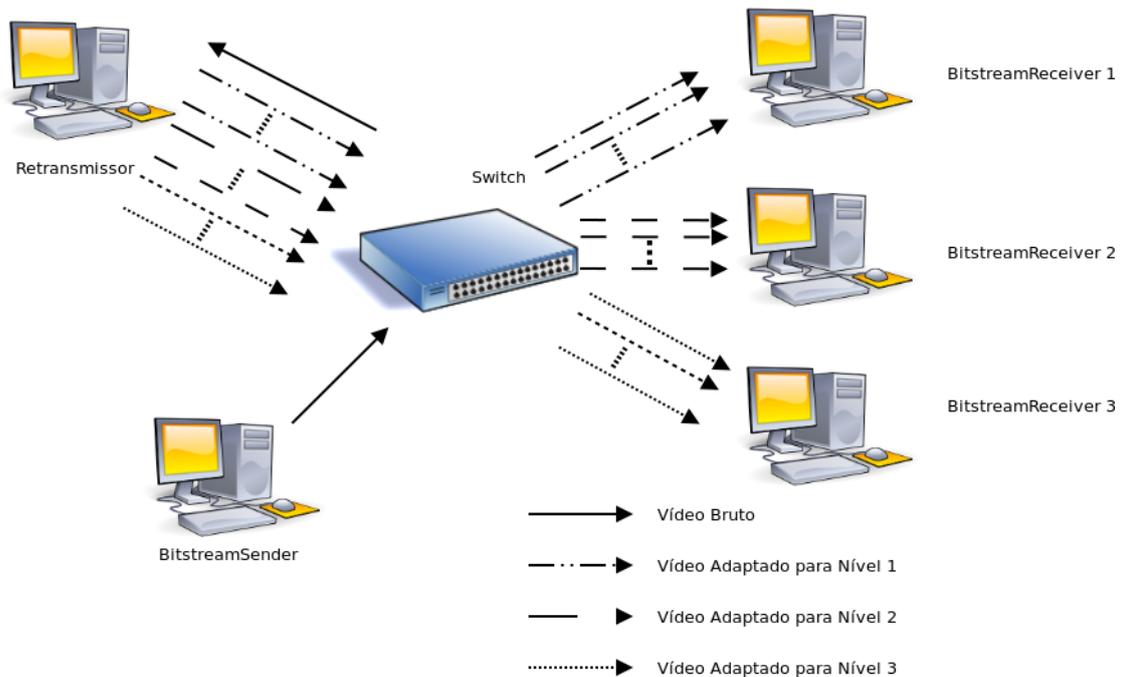


Figura 7.7: Ambiente para os testes escalabilidade do processo

O switch utilizado nos testes foi o Switch Dell PowerConnect 2824, já a configuração de cada uma das máquinas é apresentada na Tabela 7.10.

7.4.2 Resultados

Na Tabela 7.11 são apresentados os números de receptores que foi possível associar com cada um dos retransmissores, sendo que cada nível de vídeo é recebido por um terço dos receptores. Pode-se perceber, por esta tabela, que o Roteador de Vídeo Escalável se comporta muito melhor quando da presença de vários receptores. Em parte, o maior problema quanto ao método com vídeo escalável, foi a rede não comportar a taxa de bits utilizada por todos os transmissores. O processamento não foi problema para este método. O Transcodificador, por sua vez, utilizava 100% das ações das CPU's, parando de funcionar em tempo-real ainda com poucos, ou nenhum, receptores.

Tabela 7.10: Configurações das máquinas de teste

Software	Configuração da máquina
BitstreamSender	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.04 (Lucid Lynx) 32-bit
Retransmissor	Desktop Processador Intel Core 2 Quad Q9450, 2,66 GHz 2,99 Gb de Memória RAM Placa de vídeo NVidia GeForce 8600 GT Ubuntu Linux 10.10 (Maverick Meerkat) 32-bit
BitstreamReceiver 1	Desktop Dell Optiplex 960 Processador Intel Core 2 Quad Q9550 2,83GHz 3GB de memória RAM Placa de vídeo ATI Radeon HD 3470 Ubuntu Linux 10.04 (Lucid Lynx) 32-bit
BitstreamReceiver 2	Notebook Asus W5F Processador Intel Core 2 Duo T5600 1,83GHz 1GB de memória RAM Placa de vídeo Intel GMA 950 (onboard) Ubuntu Linux 10.04 (Lucid Lynx) 32-bit
BitstreamReceiver 3	Notebook Sony Vaio VGN-FE31B Intel Core 2 Duo T5500 1.66GHz 1,47 Gb de Memória RAM Placa de vídeo NVidia GeForce Go 7400 Ubuntu Linux 10.04 (Lucid Lynx) 64-bit

Tabela 7.11: Número de receptores suportados pelo retransmissor

Vídeo	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Ducks Take Off com variação de resolução	21	6
Ducks Take Off com variação de qualidade	18	3
Parkrun com variação de resolução	15	3
Parkrun com variação de qualidade	12	0
Mobcal com variação de resolução	21	6
Mobcal com variação de qualidade	12	3

8 CONCLUSÃO

O objetivo do presente trabalho foi comparar e analisar dois métodos de retransmissão adaptativa de vídeo: o transcodificador de vídeo e o roteador de vídeo escalável. Os dois métodos apresentam soluções para as necessidades da atual infra-estrutura da Internet, que ainda é muito dependente das transmissões no modelo *unicast*.

No Capítulo 4 do presente trabalho, foram apresentados os dois métodos, de acordo com os conhecimentos ilustrados nos capítulos anteriores. A partir desta base teórica, o Capítulo 6 mostrou como implementar estes dois métodos de forma simples, utilizando software livre e/ou gratuito.

Com base nos testes elaborados no Capítulo 5 e com resultados apresentados no Capítulo 7, foi possível determinar que, embora ocupe maior banda de rede para os níveis com qualidade mais alta, o Roteador de Vídeo Escalável, tem muito melhor comportamento quanto ao tempo de utilização dos processadores, além de permitir o acoplamento de muito mais receptores.

Analisando seriamente, para a maior parte das soluções, do ponto de vista do usuário, 10% ou 20% a mais de demanda de rede utilizada para a transmissão de vídeo escalável é um preço que se aceita pagar para obter vídeo com menor atraso, sem sobrecarga do retransmissor e com maior resiliência a erros. Em uma grande empresa, em que se mantém infra-estrutura de rede para video conferência com 10 Gbps de transferência entre as suas sedes, aumentar este valor para 12 Gbps, por exemplo, tem um custo baixo, para garantir transmissões com menor atraso e maior número de terminais se comunicando.

Um problema observado, também, foi a dificuldade de encontrar codificadores de vídeo escalável que codifiquem e decodifiquem o SVC do H.264 em tempo-real. A única biblioteca livre que apresenta recursos satisfatórios para utilização nos testes, ocupa, no processo de codificação e decodificação, tempos alarmantes e não chega sequer perto de atingir as necessidades de tempo-real.

As soluções propostas atualmente, compostas por um roteador de vídeo escalável, entretanto, vem-se mostrando soluções bastante eficazes para videoconferência, o que se comprova pela grande procura de grandes empresas por produtos desta natureza. Também se vislumbra a busca de algumas das maiores empresas de TI do mundo, como a Polycom, pelo desenvolvimento de soluções que utilizem o SVC do H.264 (POLYCOM, 2010).

É claro perceber que a ampliação da utilização do vídeo escalável só depende do desenvolvimento de melhores codificadores e decodificadores do SVC que possam ser utilizados mais abertamente, principalmente codificadores gratuitos ou em software livre, como os encontrados na FFMPEG. Isto fará com que o desenvolvimento de ferramentas para o SVC seja mais prático e barato.

Portanto, como trabalhos futuros, se propõe a otimização dos codificadores H.264/SVC e a utilização destes em sistemas mais robustos. Com a otimização dos codificadores, será

possível mostrar melhorias ainda mais profundas com relação ao atraso de vídeo e resiliência a erros utilizando roteadores de vídeo escalável. Alguns trabalhos neste sentido já vêm-se apresentando (HUSEMANN et al., 2009), indicando que há a possibilidade de, em breve, haver soluções em software livre de codificação de vídeo escalável em tempo-real.

Além disso, não basta que existam soluções adaptáveis, é preciso que o sistema analise automaticamente a rede, indicando qual ou quais camadas devem ser transmitidas para quais receptores. A área de avaliação e definição de parâmetros para comunicação entre os terminais permite ainda uma série de avanços que fogem do escopo deste trabalho, mas podem levar a trabalhos futuros nesta área.

A codificação de vídeo escalável vem-se provando uma área ainda incipiente, que permite margens a muita evolução. Futuramente, devem-se encontrar, de modo mais comum, soluções que utilizam o SVC do H.264, bem como outros futuros protocolos de escalabilidade de vídeo.

O trabalho se mostrou importante, pois permite avaliar as dificuldades que se apresentam para a comunicação adaptável de múltiplos pontos em redes *unicast* e comparou, cientificamente, de que forma dois modelos de retransmissão de vídeo adaptativo podem ser utilizados como solução neste contexto.

REFERÊNCIAS

AMON, P.; RATHGEN, T.; SINGER, D. File Format for Scalable Video Coding. **IEEE Trans. Circuits Syst. Video Techn.**, [S.l.], v.17, n.9, p.1174–1185, 2007.

BRUNO, G. G. E. **VEBit**: um algoritmo para codificação de vídeo com escalabilidade. 2003. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

CONKLIN, G. J.; HEMAMI, S. S. A comparison of temporal scalability techniques. **Circuits and Systems for Video Technology, IEEE Transactions**, [S.l.], v.9, n.6, p.909–919, setembro 1999.

CONNIE, A.; NASIOPOULOS, P.; LEUNG, V. C. M.; FALLAH, Y. Video Packetization Techniques for Enhancing H.264 Video Transmission over 3G Networks. In: CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE, 2008. CCNC 2008. 5TH IEEE, 2008. **Anais...** [S.l.: s.n.], 2008. p.800–804.

CÔTÉ, G.; EROL, B.; GALLANT, M.; KOSENTINI, F. H.263+: video coding at low bit rates. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.8, p.849–866, 1998.

DARONCO, L. C. **Avaliação Subjetiva de Qualidade Aplicada à Codificação de Vídeo Escalável**. 2009. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

FEIJÓO, C.; FERNÁNDEZ-BEAUMONT, J.; BARROSO, J. L. G.; MARÍN, A. Á.; ROJO-ALONSO, D. The emergence of IP interactive multimedia services and the evolution of the traditional audiovisual public service regulatory approach. **Telematics and Informatics**, [S.l.], v.24, n.4, p.272–284, 2007.

GHANBARI, M. **Standard Codecs: image compression to advanced video coding**. [S.l.]: IEE Press, 2003.

GHANDI, M. M.; GHANBARI, M. The H.264/AVC video coding standard for the next generation multimedia communications. **Journal of Iranian Association of Electrical and Electronics Engineers (IAEEE)**, [S.l.], v.1, n.2, p.3–14, 2004.

GIROD, B.; CHAKARESKI, J.; KALMAN, M.; LIANG, Y. J.; SETTON, E.; ZHANG, R. Advances in Network-adaptive Video Streaming. In: PROC. OF THE INTERNATIONAL WORKSHOP ON DIGITAL COMMUNICATIONS (IWDC 2002, 2002. **Anais...** [S.l.: s.n.], 2002. p.1–8.

- GIROD, B.; KALMAN, M.; LIANG, Y. J.; ZHANG, R. Advances in Channel-adaptive Video Streaming. **Wireless Communications and Mobile Computing**, [S.l.], v.2, p.549–552, 2002.
- HASKELL, B. G.; PURI, A.; NETRAVALI, A. N. **Digital Video: an introduction to mpeg-2**. [S.l.]: Springer US, 1997. p.183–229.
- HEI, X.; LIANG, C.; LIANG, J.; LIU, Y.; ROSS, K. W. A measurement study of a large-scale P2P IPTV system. **Multimedia, IEEE Transactions on**, [S.l.], 2007.
- HSIEH, J.-W.; LIN, M.; LIU, J. C. L.; DU, D. H.-C.; RUWART, T. Performance of a Mass-Storage System for Video-on-Demand. **J. Parallel Distrib. Comput.**, [S.l.], v.30, n.2, p.147–167, 1995.
- HÜSEMANN, R. **Arquitetura em hardware para implementação de um codificador de vídeo escalável padrão SVC**. 2008. Proposta de Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul.
- HUSEMANN, R.; DETSCH, A.; ROESLER, V.; LIMA, J. V.; SUSIN, A. A. Practical analysis and refinement of a SVC standard scalable video coder. In: XV BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB, 2009, New York, NY, USA. **Proceedings...** ACM, 2009. p.27:1–27:8. (WebMedia '09).
- ITU-T. **Recommendation ITU-T H.264**. [S.l.: s.n.], 2010.
- JACK, K. **Video Demystified: a handbook for the digital engineer**. 5a.ed. [S.l.]: Newnes, 2007. 976p.
- KOENEN, R. **Overview of the MPEG-4 Standard**. Geneva, Switzerland: International Organization for Standardization, ISO/IEC JTC1/SC29/WG11 N2323, Coding of Moving Pictures and Audio, 2002.
- KONDI, L. P.; ISHTIAQ, F.; KATSAGGELOS, A. K. On video SNR scalability. **Image Processing, International Conference on**, Los Alamitos, CA, USA, v.3, p.934, 1998.
- LEWIS, B.; BERG, D. J. **Multithreaded programming with Pthreads**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.
- LI, H.; LI, Z. G.; WEN, C. Fast Mode Decision Algorithm for Inter-Frame Coding in Fully Scalable Video Coding. **Circuits and Systems for Video Technology, IEEE Transactions on**, [S.l.], v.16, n.7, p.889–895, 2006.
- LIN, J.-L.; HWANG, W.-L.; PEI, S.-C. SNR scalability based on bitplane coding of matching pursuit atoms at low bit rates: fine-grained and two-layer. **IEEE Trans. Circuits Syst. Video Techn.**, [S.l.], v.15, n.1, p.3–14, 2005.
- LIU, J.; LI, B.; ZHANG, Y.-Q. Optimal stream replication for video simulcasting. **IEEE Transactions on Multimedia**, [S.l.], v.8, n.1, p.162–169, 2006.
- MANSOUR, H.; FALLAH, Y.; NASIOPOULOS, P.; KRISHNAMURTHY, V. Dynamic Resource Allocation for MGS H.264/AVC Video Transmission Over Link-Adaptive Networks. , [S.l.], v.11, n.8, p.1478–1491, dezembro 2009.

MASSARENTI JR., N. D. Utilização de Softwares Livres em Educação a Distância em Medicina e Saúde: uma experiência de 6 anos. In: CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE, 2006, Florianópolis, Santa Catarina. **Anais...** Sociedade Brasileira de Informática em Saúde, 2006. p.5.

NEUMANN, C.; ROCA, V. **Scalable Video Streaming over ALC (SVSoA): a Solution for the Large Scale Multicast Distribution of Videos**. [S.l.]: INRIA, 2003. 0. (RR-4769).

OELBAUM, T.; SCHWARZ, H.; WIEN, M. Performance Analysis of SVC. **IEEE Trans. Circuits and Systems for Video Technology**, [S.l.], v.17, n.12, p.1771–1771, dezembro 2007.

OSTERMANN, J.; BORMANS, J.; LIST, P.; MARPE, D.; NARROSCHKE, M.; PEREIRA, F.; STOCKHAMMER, T.; WEDI, T. Video coding with H.264/AVC: tools, performance, and complexity. **Circuits and Systems Magazine, IEEE**, [S.l.], v.4, n.1, p.7–28, 2004.

POLYCOM. **More Scale at Lower Cost with Scalable Video Coding**. 2010.

RICHARDSON, I. E. **H.264 and MPEG-4 Video Compression Video Coding for Next Generation Multimedia**. 1.ed. [S.l.]: Wiley, 2003.

ROBERS, M. A. **SNR Scalable Video Coder Using Progressive Transmission of DCT Coefficients**. 1997.

ROESLER, V.; HUSEMANN, R.; COSTA, C. H. A new multimedia synchronous distance learning system: the iva study case. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2009., 2009, New York, NY, USA. **Proceedings...** ACM, 2009. p.1765–1770. (SAC '09).

SCHWARZ, H.; HINZ, T.; MARPE, D.; WIEGAND, T. **Technical description of the HHI proposal for SVC CE2**. Palma de Mallorca, Spain: MPEG ISO/IEC, 2004.

SCHWARZ, H.; MARPE, D.; SCHIERL, T.; WIEGAND, T. Combined scalability support for the scalable extension of H.264/AVC. In: ICME, 2005. **Anais...** IEEE, 2005. p.446–449.

SCHWARZ, H.; MARPE, D.; WIEGAND, T. Analysis of Hierarchical B Pictures and MCTF. **Multimedia and Expo, IEEE International Conference on**, Los Alamitos, CA, USA, v.0, p.1929–1932, 2006.

SCHWARZ, H.; MARPE, D.; WIEGAND, T. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. **Circuits and Systems for Video Technology, IEEE Transactions on**, [S.l.], v.17, n.9, p.1103–1120, 2007.

SOLER, L. **Compactação de Vídeo Escalável**. 2006. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

SULLIVAN, G. J.; TOPIWALA, P. N.; LUTHRA, A. The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions. In: APPLICATIONS OF DIGITAL IMAGE PROCESSING XXVII, 2004. **Anais...** SPIE, 2004. v.5558, n.1, p.454–474.

TAN, W. tian; ZAKHOR, A. Video Multicast using Layered FEC and Scalable Compression. **IEEE Trans. Circuits Syst. Video Technol**, [S.l.], v.11, p.373–386, 2001.

THOM, G. A. H.323: the multimedia communications standard for local area networks. **Communications Magazine, IEEE**, [S.l.], v.34, n.12, p.52–56, 1996.

TUDOR, P. MPEG-2 video compression. **Electronics Communication Engineering Journal**, [S.l.], v.7, n.6, p.257 – 264, Dec. 1995.

VETRO, A.; CHRISTOPOULOS, C.; SUN, H. Video transcoding architectures and techniques: an overview. **IEEE Signal Processing Magazine**, [S.l.], v.20, n.2, p.18–29, março 2003.

WANG, Y.-K.; HANNUKSELA, M.; PATEUX, S.; ELEFThERiADiS, A.; WENGER, S. System and Transport Interface of SVC. **Circuits and Systems for Video Technology, IEEE Transactions on**, [S.l.], v.17, n.9, p.1149–1163, 2007.

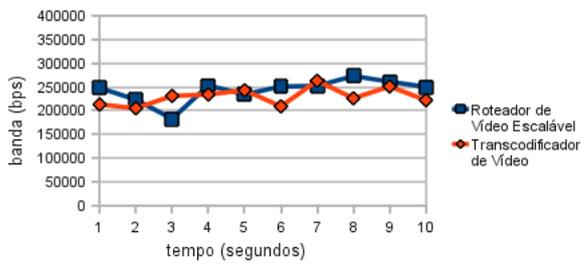
WEBSTER, A. A.; JONES, C. T.; PINSON, M. H.; VORAN, S. D.; WOLF, S. An Objective Video Quality Assessment System Based on Human Perception. In: SPIE HUMAN VISION, VISUAL PROCESSING, AND DIGITAL DISPLAY IV, 1993. **Anais...** [S.l.: s.n.], 1993. p.15–26.

WENGER, S. H.264/AVC over IP. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.13, n.7, p.645–656, julho 2003.

WIEGAND, T.; SULLIVAN, G. J.; BJONTEGAARD, G.; LUTHRA, A. Overview of the H.264/AVC video coding standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.13, n.7, p.560–576, julho 2003.

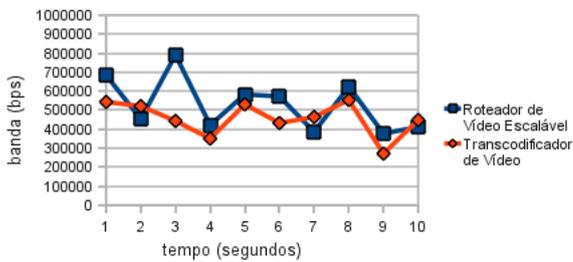
APÊNDICE A RESULTADOS DOS TESTES DE DEMANDA DE REDE

Nível 1



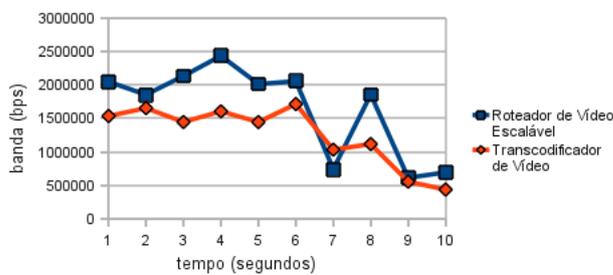
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	237.109	225.292

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	517.578	444.824

Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Bitrate (kbps)	1604.003	1222.949

Figura A.1: Demanda de rede nos três níveis do vídeo Parkrun com resolução variando

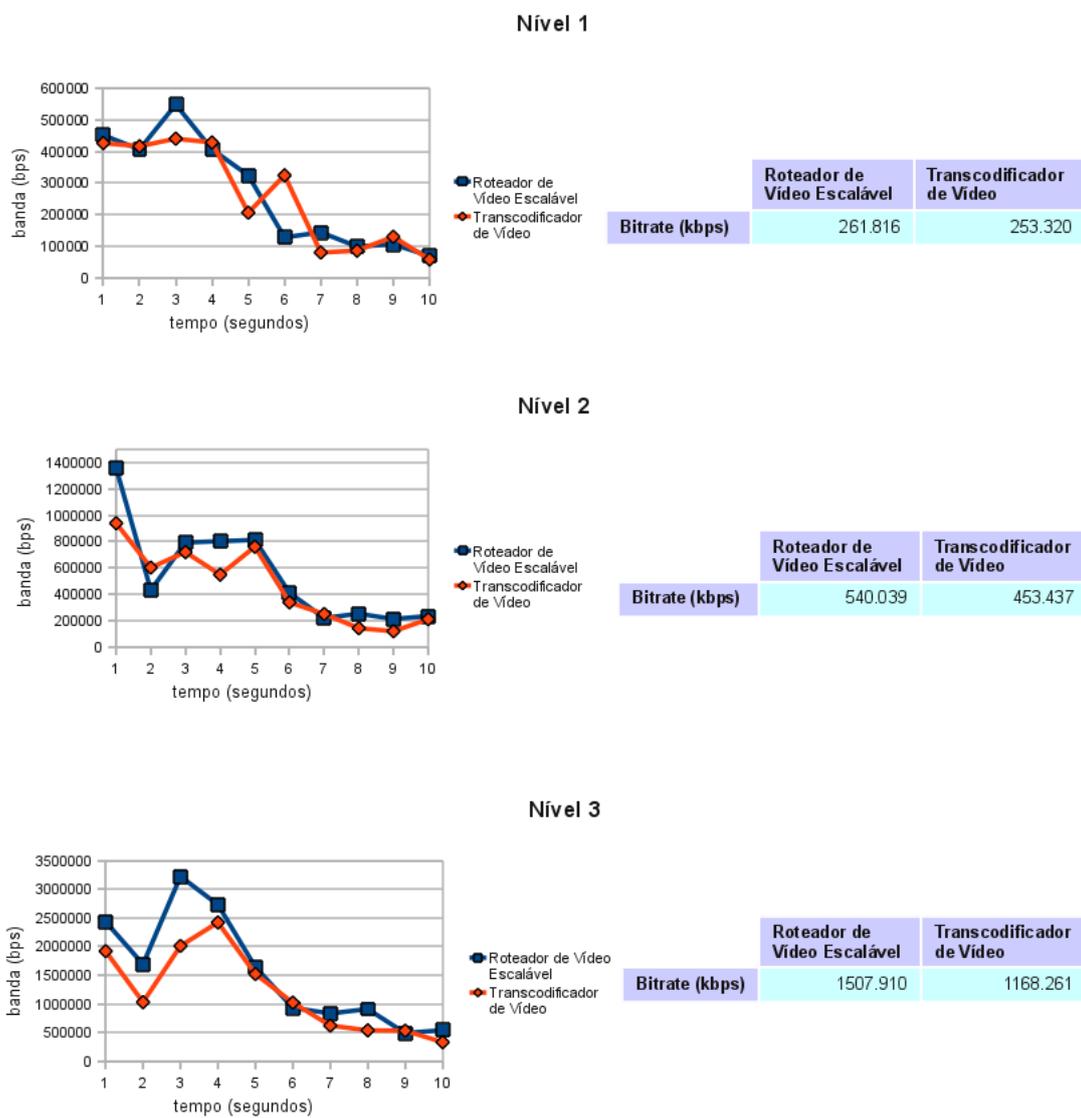


Figura A.2: Demanda de rede nos três níveis do vídeo Parkrun com qualidade variando

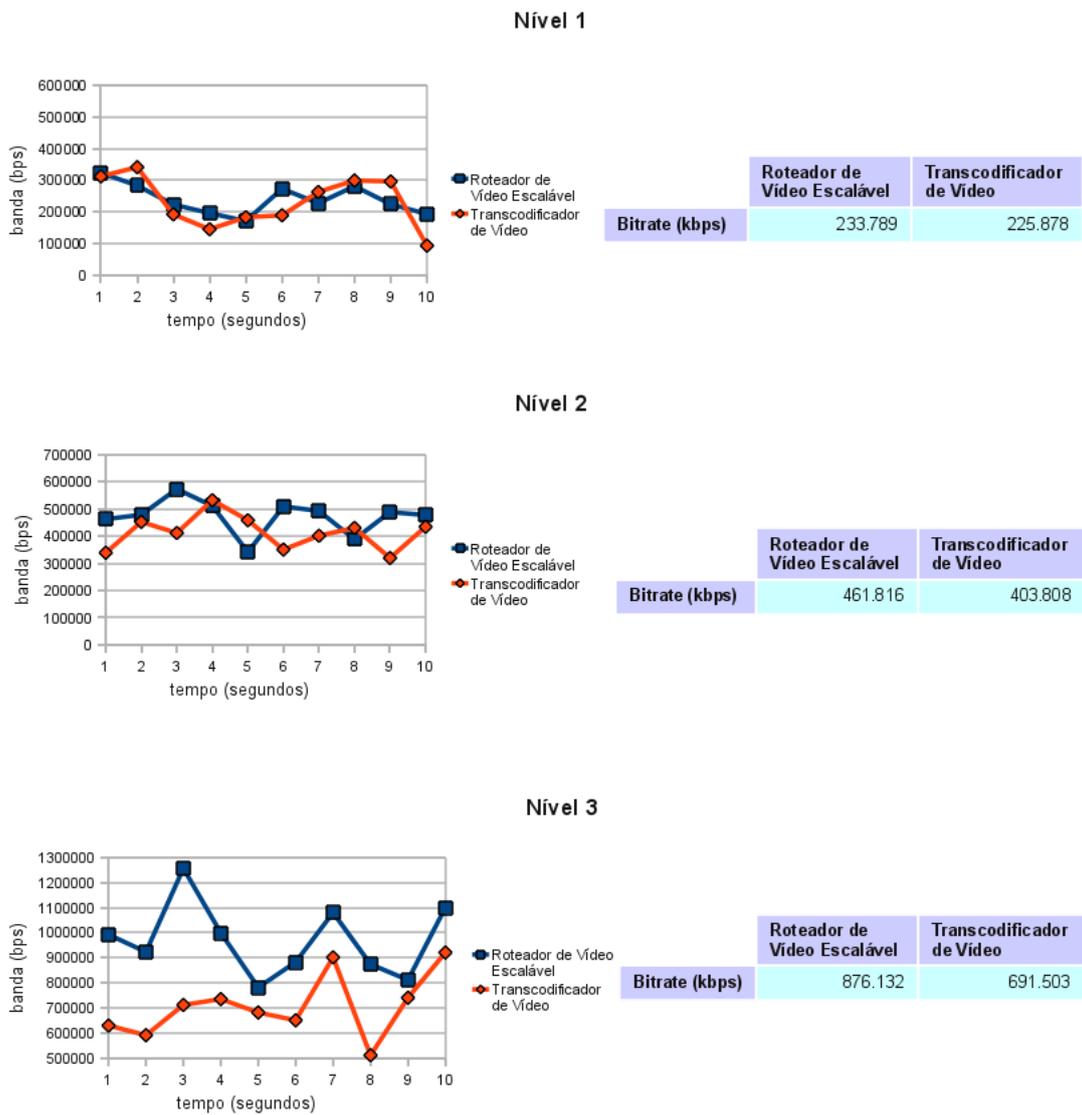


Figura A.3: Demanda de rede nos três níveis do vídeo Mobcal com resolução variando

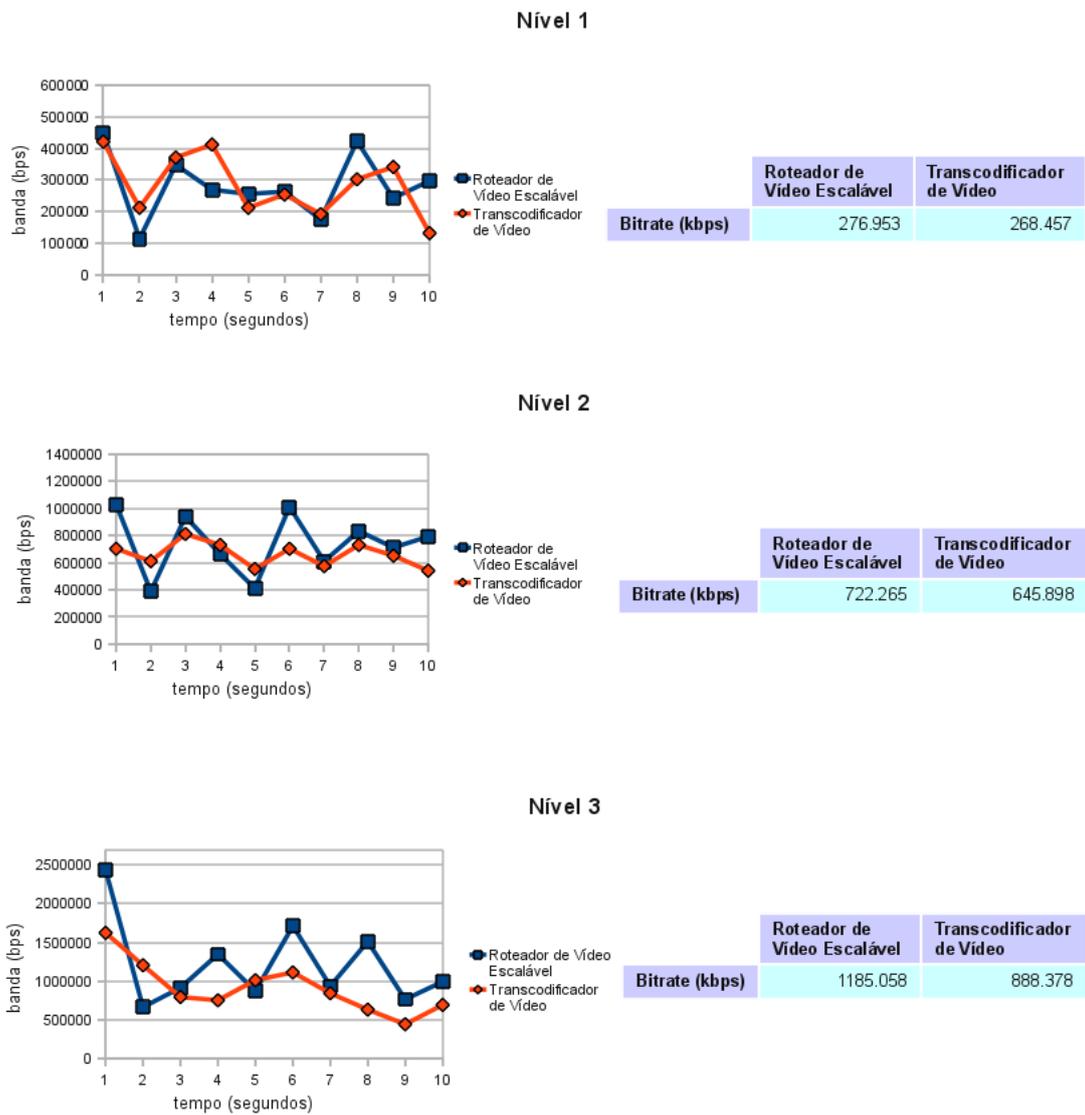
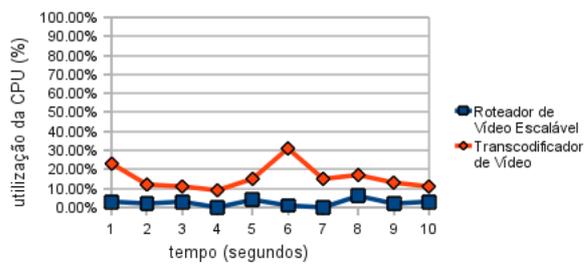


Figura A.4: Demanda de rede nos três níveis do vídeo Mobcal com qualidade variando

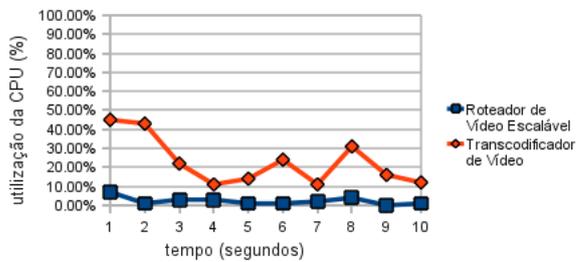
APÊNDICE B RESULTADOS DOS TESTES DE DEMANDA COMPUTACIONAL

Nível 1



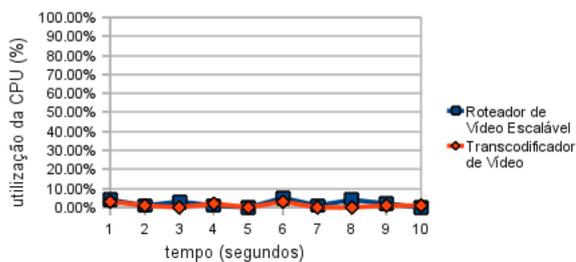
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	2.4%	15.6%

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	2,3%	22.8%

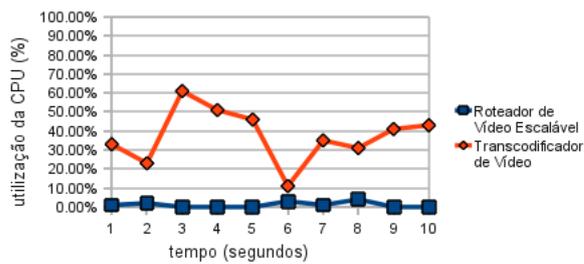
Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	2.1%	1.1%

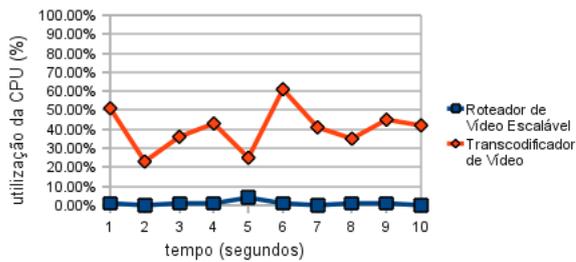
Figura B.1: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Parkrun com resolução variando

Nível 1



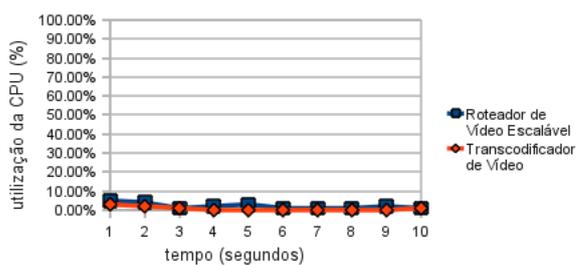
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1.1%	37.5%

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1,0%	40,2%

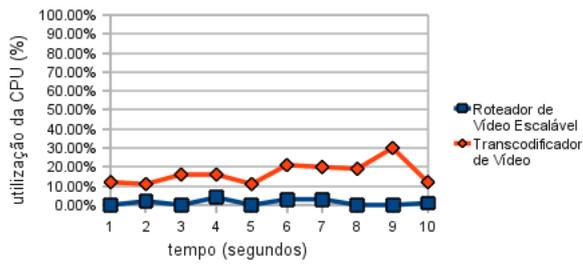
Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	2,1%	0,7%

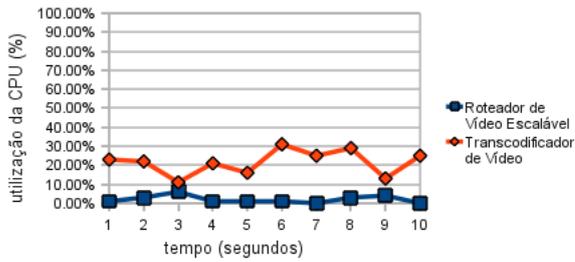
Figura B.2: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Parkrun com qualidade variando

Nível 1



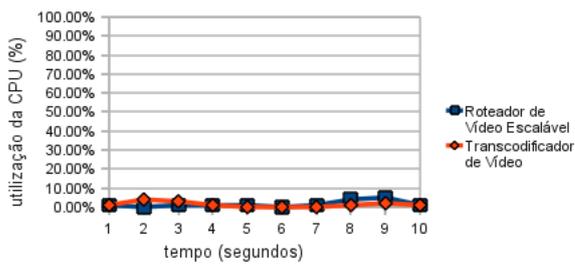
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1.3%	16.8%

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	2,0%	21,6%

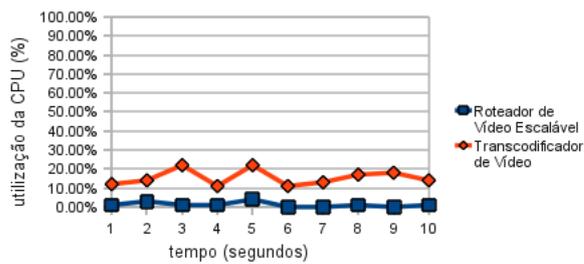
Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1,5%	1,3%

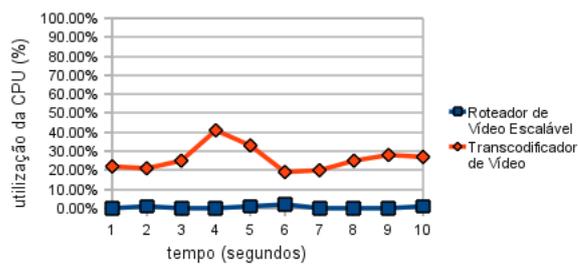
Figura B.3: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Mobcal com resolução variando

Nível 1



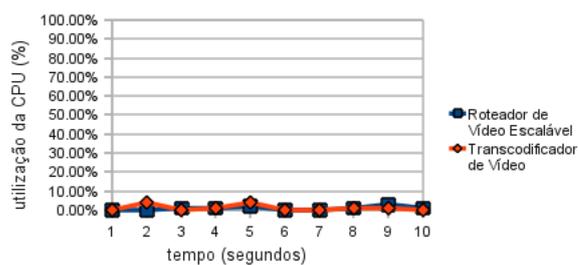
	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	1,2%	15,4%

Nível 2



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	0,5%	26,1%

Nível 3



	Roteador de Vídeo Escalável	Transcodificador de Vídeo
Média de Processamento (%)	0,9%	1,1%

Figura B.4: Porcentagem de utilização da CPU na retransmissão dos três níveis do vídeo Mobcal com qualidade variando